Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Materials & Chemical Sciences Division

National Center for Electron Microscopy

Presented at the 6th Pfefferkorn Conference,
Niagara Falls, NY, April 28–May 2, 1988
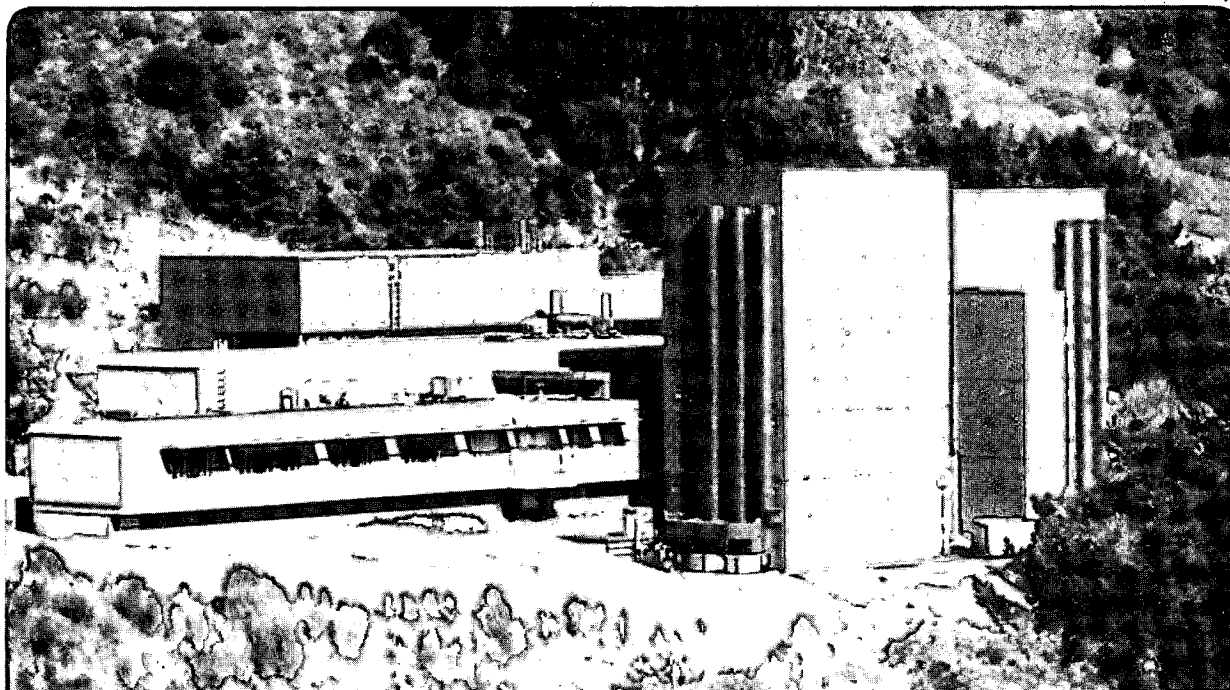
Advances in High-Resolution Image Simulation

M.A. O'Keefe and R. Kilaas

September 1988

DISCLAIMER

ADVANCES IN HIGH-RESOLUTION IMAGE SIMULATION

Michael A. O'Keefe & Roar Kilaas

National Center for Electron Microscopy
Building 72, Lawrence Berkeley Laboratory
1 Cyclotron Road
Berkeley, CA 94720

Telephone (415) 486 4610

Abstract

Continuing advances in hardware and software have improved both the speed and the range of computations that can be made to simulate high resolution electron microscope (HREM) images from various structures. Use of image display systems and array processors have made the image simulation procedure much more interactive while laser printers provide a fast high-quality hard copy output.

Use of array processors has enabled the rewriting of electron scattering algorithms to include convergence effects (previously only considered after the scattered electron beams had emerged from the specimen) and upper-layer-line effects. With an array processor it is faster to compute the effects of spatial and temporal coherence in real space, rather than use approximate solutions derived from series expansion in reciprocal space.

With a frame buffer and suitable software the user has the facility to change parameters and view the results of the change almost immediately. Selected images can then be directed to hard copy output, in contrast with batch methods where series of hard copy images are produced and then selected from. Given a microdensitometer for input of experimental images from plates, or a video camera attached to the electron microscope and a frame buffer, split screen comparisons between experimental and computed images are possible, including independent control of image contrast, magnification and orientation.

Introduction

Since the earlier report by O'Keefe (1984), improvements in both hardware and software have made possible improvements in HREM image processing. Recent advances in simulation (and also processing) of high-resolution electron images fall mainly into two broad categories, improvements in the ease of use of the programs, and improvements in the accuracy and power of the algorithms used to describe the physical processes. In both categories the on-going evolution of computer hardware has prompted, or at least facilitated, these improvements. Up to a few years ago, image simulations of high resolution electron micrographs were largely done on large mainframe computers and often overnight. Images were printed out the next day in the form of overprinted characters on large-format lineprinter paper. Today, it is possible to take a micro/mini-computer such as a MicroVax or a Sun, connect it to a small array processor (and possibly a framestore) and compute and display images faster than is possible on a multi-user mainframe computer. Use of a laser printer for fast hardcopy output (e.g. Ackland et al., 1986) provides a much-improved image over overprinted character display. The existence of this type of hardware has increased demand for interactive software (Kilaas, 1987a), increased the size of the computation that can be handled in reasonable time (Kilaas, 1987b), and increased the level of sophistication that can be reached in describing the electron scattering process and the electron microscope imaging system (Kilaas et al., 1987).

Much of the improvement in the "number-crunching" aspect of the simulation process is linked to the availability of low-cost array processors and is discussed in terms of these devices. Similarly, many improvements in the software "user interface" are dependent on the use of a video data terminal (VDT) for data input and a graphics terminal or frame buffer for image display; these interactive improvements are also discussed. Of course, the two groups of improvements are not independent; the large increase possible in the speed of the diffraction part of the image computation tends to make input of the model structure the time-consuming part of the process, and places many incentives on ways to facilitate the input of data needed to describe the model structure.

Improvements in "Number-Crunching"

The processing power of array processors is generally quoted in millions of floating-point operations per second (MFLOPS). The advent of small cheap array processors (like the 20 MFLOPS MCS ZIP3232, the 15 MFLOPS SKY Warrior, the 20 MFLOPS CSA uMSP-4 and the 32 MFLOPS CSPI QuickCard) has enabled an increase in both the size and the complexity of electron image simulations, particularly in diffraction computations using the multislice technique of Goodman and Moodie (1974). Early simulation programs like SHRLI80 (O'Keefe et al., 1978) used 128x128 sampling points (parameter-14 arrays) to compute the diffraction of up to 4000 beams. Ten years ago such a computation required over 400 seconds per specimen "slice" on a mainframe (Univac 1108) computer (O'Keefe and Iijima, 1978); today a uVAXII minicomputer takes 18 seconds, and a uVAX with a CSPI MM+4 array processor requires only 0.2 seconds; at these speeds, the computation approaches "real time" and begs for an interactive format. Larger computations become possible; whereas a parameter-14 multislice (128x128 sampling points) is large enough for accurate computation from (projected) unit cells (or defect cells) up to areas of 2.5 nm^2 for thick crystals (and 10 nm^2 for very thin crystals), a parameter-18 multislice (array sizes of 512x512 points, or 1024x256 points, ..., or 8192x32 points, or 16384x16 points) can accommodate cells up to 40 nm^2 in area (even for thick crystals) and takes only 4 seconds per slice with a 20MFLOPS array processor.

FFT Multislice and the Array Processor

As introduced by Goodman and Moodie (1974), the basic recursive form of the multislice description of dynamical diffraction can be written as

$$\Psi_{n+1}(k) = [\Psi_n(k) \cdot P_{n+1}(k)] * Q_{n+1}(k) \qquad (1)$$

That is, $\Psi_{n+1}(k)$, the wave function (given in reciprocal space) at the exit surface of the (n+1)th slice, is obtained by multiplying the wave function at the exit surface of the nth slice by $P_{n+1}(k)$, the propagator for the (n+1)th slice, followed by the convolution of this result by $Q_{n+1}(k)$, the phase-grating function (given in reciprocal space) for the (n+1)th slice.

In multislice computer programs that use the above reciprocal-space formulation, the three functions, $\Psi(k)$, $P(k)$, and $Q(k)$ are represented by two-dimensional arrays containing terms with indices that can be regarded as those of the diffracted beams within the diffracting specimen. In order to impose no extra symmetry on the computation, a "circular aperture" is usually placed on the terms in the arrays, so that terms beyond a certain distance in reciprocal space are set to zero. Note that the $Q(k)$ array must extend out to twice as far in reciprocal space as the $\Psi(k)$ and $P(k)$ arrays in order to correctly include all physical scatterings to each diffracted beam. Figure 1 shows how this doubled phase-grating requirement arises because $Q(k)$ is a "probability map" of terms that determine how much of each diffracted beam is to be scattered through the angle corresponding to the term. Thus, in order to compute the scattering of diffracted beams out to $k = \pm(h,k)$, the $\Psi(k)$ and $P(k)$ must be $(2h+1)x(2k+1)$ in size, and $Q(k)$ must be $(4h+1)x(4k+1)$. For example, the SHRLI80 programs use a 128x128 array for $Q(k)$, giving a maximum (h,k) value of (31,31); this limits the number of diffracted beams to less than 4096, adequate for most perfect-crystal computations, but too small for simulation of many defect structures.

For larger arrays (and thus larger numbers of diffracted beams), it is faster to compute the multislice by recasting (1) into a form where the convolution is replaced by two Fourier transforms and a multiply operation, as was done by early users of multislice at Melbourne (e.g. O'Keefe, 1972). Then

$$\Psi_{n+1}(k) = F[F^{-1}[\Psi_n(k) \cdot P_{n+1}(k)] \cdot q_{n+1}(x)] \qquad (2)$$

Here the F represents the Fourier transform operation, and q(x) is the real-space form of Q(k), i.e. the (inverse) Fourier transform of Q(k). Experience with the SHRLI programs has produced the not-surprising result that, in order to obtain the same results with (2) as with (1), q(x) must be formed from a Q(k) that extends twice as far in reciprocal space as do $\Psi(k)$ and $P(k)$. Note that, as well as correctly describing the physics of the scattering process by ensuring that all contributions from all the diffracted beams leaving the previous slice to each diffracted beam from the current slice are included, the doubled phase-grating also neatly eliminates the possibility of any "aliasing" arising from the necessary sampling of the real-space functions. Unfortunately, in their description of the FFT multislice, Ishizuka and Uyeda (1977) stated that the number of diffracted beams must be no less than the number of phase-grating coefficients; under this condition, the effects of aliasing will produce incorrect scattering results if array sizes are not made large enough to produce essentially zero electron amplitude over the outer five-nineths of the phase-grating and wave-amplitude arrays (see discussion with the referee at the end of this report).

For a $Q(k)$ array of 128x128, the FFT multislice of equation 2 requires the following recursion steps:

1. multiply (a 64x64) $\Psi_n(k)$ by (a 64x64) $P_{n+1}(k)$.
2. (inverse) Fourier transform the result of (1) embedded in a 128x128 array.
3. multiply the result of (2) by (a 128x128) $q_{n+1}(x)$ formed from (a 128x128) $Q(k)$.
4. FFT transform the result of (3); set all values outside the inner 64x64 to zero.

Of course the arrays need not be square, although it is convenient to make their dimensions equal to powers of 2 to suit most available fast Fourier transform (FFT) routines. Then, for a $2^n \times 2^m$ $Q(k)$ array (or the same size $q(x)$ array), the computation of $2^{n-1} \times 2^{m-1}$ diffracted beams requires, for every slice of the multislice computation, two $2^n \times 2^m$ FFTS plus $1.25 \times 2^n \times 2^m$ (complex) multiply operations.

In an array processor, a two-dimensional FFT is typically carried out as a series of one-dimensional FFTs; the rows are transformed first, followed by transformation of the columns. For a $2^n \times 2^m$ array, this process requires 2^n FFTs of size 2^m followed by 2^m FFTs of size 2^n. Thus, one multislice recursion would require $2 \times [2^n \times (2^m FFT) + 2^m \times (2^n FFT)]$ operations plus $(1.25 \times 2^n \times 2^m)$ multiply operations, if the steps (1) through (4) above were used.

In programming an FFT multislice algorithm on an array processor, it is possible to take advantage of the reduced size of $\Psi(k)$ and $P(k)$ (compared with $Q(k)$) to reduce the number of one-dimensional FFTs required. Figure 2 shows the steps involved for a $2^n \times 2^m$ multislice; these steps apply to all array processors, including those able to fit only one single row or column in memory at one time, and those capable of containing all of the required arrays simultaneously.

Starting at step (1), we clear (set to zero) the top and bottom rows outside the inner 2^{n-1} rows. We then loop over the remaining rows; for each row, we FFT it (step 2) to bring it to reciprocal space, then (step 3) clear the ends of the row to leave 2^{m-1} terms; after multiplying the row terms (step 4) by 2^{m-1} terms of a row of the propagator $P(k)$, we perform an inverse FFT on the row (step 5) to bring it halfway to real space. After looping over all rows (actually just the inner one-half since the rest are zero), we loop over the columns. For each column we perform an inverse FFT (step 7) to bring it fully to real space, then multiply it (step 8) by a 2^n column of $q(x)$, and perform a forward FFT (step 9) to bring the column halfway to reciprocal space. When all columns are completed (step 10), we clear the top and bottom rows again (step 1).

For array processors able to hold only one row or column in on-board memory at one time, this procedure minimizes the number of slow (host) memory to fast (on-board) memory transfers required; for array processors with sufficient memory to hold all three arrays [$\Psi(k)$, $P(k)$, and $q(x)$], no transfers are necessary (as long as all slices are the same) and the computation is accordingly faster. In both cases the number of operations required for computation is $2 \times [2^{n-1} \times (2^m FFT) + 2^m \times (2^n FFT)]$ FFT operations plus $2^{n-1}(2^{m-1}) + 2^m(2^n)$ multiply operations, thus cutting the number of FFT operations by 25 percent over the algorithm using standard two-dimensional FFTs, without increasing the number of multiply operations.

In order to obtain output from the procedure, rows of $\Psi(k)$ can be taken out after step (3), or columns of $\psi(x)$ can be removed after step (7). For input to start the procedure, rows of $Q_1(k)$ can be input at step (3), or columns of $q_1(x)$ can be input at step (8).

A third method of computing the multislice is to carry out the calculation completely in real space; this is an alternative to computing in reciprocal space (as

in equation 1), or in both real and reciprocal space (as in equation 2). Fourier transformation of (1) gives

$$\psi_{n+1}(x) = [\psi_n(x) * p_{n+1}(x)] \cdot q_{n+1}(x) \qquad (3)$$

In this formulation the real-space wave function $\psi_n(x)$ is convolved by a real-space propagator $p_{n+1}(x)$ and the result multiplied by the real-space phase-grating. Van Dyck (1983) has developed this method and shown that the convolution step need not be carried out with the full $2^n \times 2^m$ p(x) array, but only with a such smaller array (since p(x) is sharply peaked in the forward direction for high energy scattering). Since the modified p(x) array is so much smaller, the computation time should take less time than either of the reciprocal-space multislice (1) or the FFT multislice (2); however, tests (e.g. Kilaas and Gronsky, 1983) have shown that the original formulation of the method required more beams and smaller slices to achieve the same precision as either the reciprocal-space multislice or the FFT multislice, resulting in a longer time to produce the wavefield at the same total specimen thickness. Van Dyck and Coene (1984) have since proposed a modified implementation of the real-space multislice into a workable algorithm, with results that approach those produced by the multislice formulations of equations 1 and 2 (Coene and Van Dyck, 1984). Since the p(x) array is much smaller, procedure (3) does produce some saving in memory over the other multislice formulations.

The "size" of a multislice calculation is defined conveniently by the size of the largest array (the Q(k) or q(x) array). For an array size of $2^n \times 2^m$, the multislice is said to be a "parameter n+m" multislice; thus a parameter-14 multislice will have a Q(k) array size of 128x128, or 256x64, or 512x32, or 1024x16 depending upon the shape of the unit cell used in the computation.

All three formulations of the multislice procedure require the computer (or array processor) to store three different complex arrays corresponding to the phasegrating function in real or reciprocal space (q(x) or Q(k)), the electron wavefield in real or reciprocal space ($\psi(x)$ or $\Psi(k)$), and the propagator function in real or reciprocal space (p(x) or P(k)). Table 1 compares the memory requirements of the three multislice formulations.

With a reciprocal space phasegrating Q(k), of $2^n \times 2^m$ terms, a reciprocal-space multislice formulation (equation 1), requires a reciprocal-space wavefield array (or set of diffracted beams), $\Psi(k)$, of $2^{n-1} \times 2^{m-1}$ terms, and a reciprocal-space propagator array P(k), of $2^{n-1} \times 2^{m-1}$ terms.

An equivalent FFT multislice (equation 2) requires a real-space phase-grating array q(x), of $2^n \times 2^m$ terms, an electron wavefield array $\Psi(k)$, of $2^n \times 2^m$ terms, and a reciprocal-space propagator array P(k), of $2^{n-1} \times 2^{m-1}$ terms.

For a real-space multislice (equation 3), both the $\psi(x)$ and q(x) arrays need to hold $2^n \times 2^m$ terms, whereas p(x) may be as small as thirteen (Kilaas & Gronsky, 1983), giving a total requirement of slightly over $2 \times 2^{n+m}$ terms. In implementing the real-space multislice it is important that the $2^n \times 2^m$ q(x) array should be formed from a full $2^n \times 2^m$ Q(k) terms, whereas the $2^n \times 2^m$ $\psi(x)$ array should be formed from only $2^{n-1} \times 2^{m-1}$ $\Psi(k)$ terms in order to include correctly all the physical scattering contributions to each diffracted beam and to avoid aliasing problems.

Any (correctly implemented) multislice of size $2^n \times 2^m$ (parameter n+m) thus includes the effects of only $2^{n-1} \times 2^{m-1}$ diffracted beams; this is true for all three formulations of the multislice: the reciprocal space method (equation 1), the FFT method (equation 2), and the real-space multislice (equation 3).

HOLZ Interactions

With array processors it is possible to include in the diffraction calculation the effects of out-of-zone scatterings, or non-zero (or higher-order) Laue zone (HOLZ) interactions, and yet remain within a reasonable time limit.

A multislice computation in which every slice is identical contains no information about the variation in structure along the incident beam direction, and includes scattering interactions with only the zero-order Laue zone (ZOLZ) layers. For structures with short repeat distances in the beam direction such a computation is adequate, since the Ewald sphere will not approach high-order zones.

For structures with large repeats in the beam direction, several methods may be used, three of which rely on sub-dividing the slice in order to compute the electron scattering. The simplest, but most approximate method, is to compute the projected potential for the full repeat period then use 1/n of the projected potential to form a phase-grating function that can be applied n times to complete the slice. This method avoids interaction with any "pseudo-upper-layer-line" (Goodman and Moodie, 1974), but ignores real HOLZ layers.

An improvement on sub-dividing the projected potential is to sub-divide the unit cell atom positions. In this procedure the list of atom positions within the unit cell is divided into n sub-slices perpendicular to the incident beam direction. From these sub-slices, different projected potentials are produced to form n different phase-gratings, which are applied successively to produce the scattering from the full slice.

A further improvement on sub-dividing the atom positions, is to sub-divide the three-dimensional potential of the full slice, since an atom with a position within one sub-slice can have a potential field that extends into the next sub-slice. Rather than compute a full three-dimensional potential and then integrate over appropriate sub-slices (a 128x128x128 potential would require over two million samples to be stored), it is possible to derive an analytical expression for the potential within the sub-slice $z_0 \pm \Delta z$ projected onto the plane at z_0 (Self et al., 1983). With an array processor it is possible to apply this method routinely to structures with large repeats in the beam direction, thus generating several different phase-gratings for successive application, and even to structures (perhaps with defects) that are aperiodic in the beam direction and require a large number of individual non-repeating phase-gratings (Kilaas et al., 1987).

Van Dyck has proposed other methods to include the effects of HOLZ layers, including the second-order multislice with potential eccentricity (Van Dyck, 1980) and the improved phase-grating method (Van Dyck, 1983). Tests of these procedures show that the extra computation involved in using potential eccentricity is suited to use of an array processor, but that the improved phase-grating method diverges too easily to be useful.

Partial Coherence and the Array Processor

The spatial frequencies transferred into the image intensity spectrum from the image amplitude spectrum (i.e. from the set of diffracted beams that pass through the objective aperture) are strongly influenced by the extent of spatial and temporal coherence of the diffracted beams emerging from the specimen. These coherences, in fact, determine the overall resolution of the electron microscope (O'Keefe, 1979). Early simulation programs incorporated spatial coherence effects by using a grid to sample the emergent cones of diffracted beams, producing an image with the correct misalignment (or beam tilt) from the objective axis for each sampling angle, then summing the image intensities (O'Keefe and Sanders, 1975). While this approach produced images that matched experimental ones, it provided no insight into the

resolution-worsening effect of the convergent incident beam. Following Frank's (1973) formulation of the convergence and spread-of-focus effects (i.e. the effects of partial spatial and temporal coherence), later programs (O'Keefe et al., 1978) used a reciprocal-space formulation to include the effects in simulated images by performing a double sum over all the beams admitted by the objective aperture, modulating each contributing term by a suitable cross-coefficient damping term (O'Keefe, 1979).

As electron microscope resolutions improve, more diffracted beams are admitted by the objective aperture to contribute to the image. Similarly, the use of large defect "cells" generates a correspondingly large number of terms in the amplitude spectrum of the image. Both of these factors increase the time required for the application of coherence effects by the reciprocal-space double-sum method. With a 15 to 20 MFLOPS array processor, the real-space sum-over-intensities method becomes faster than the reciprocal-space double-sum method once the number of amplitude spectrum terms (i.e. diffracted beam amplitudes passed by the objective aperture) becomes greater than 50 to 100, a situation that is easily attained with most defect simulations. For these reasons the real-space sum-over-intensities method of incorporating the effects of partial spatial (and temporal) coherence is the preferred one when an array processor is available.

Figure 3 illustrates the method. Figure 3a shows an experimental diffraction pattern obtained with focussed illumination, and with the objective aperture superimposed. Figure 3b shows how each diffraction disk is sampled (in this case in 49 equal-area segments) in order to calculate an image at the misalignment angle corresponding to each sampling point. An additional advantage of the real-space sum-of-intensities method over the reciprocal-space double-sum method is the ease of handling the intersections of diffracted cones with the objective aperture. Figure 3 shows how the outer beam cones are truncated by the aperture. In the real-space method this situation is modeled accurately by simply excluding those sampling points that lie outside the aperture. In the reciprocal-space method the outer disks are weighted by the area falling inside the aperture and then treated like those that fall completely inside. Since the objective lens phase change across these outer beams is large, this procedure is only approximate. In the case of dark-field images, where only a small number of beams contribute to the image, the number of intersected cones may exceed the number of complete cones (figure 4), making the correct handling of the intersections essential (Iijima and O'Keefe, 1977).

The sampling points in Figure 3b were chosen to provide sampling intervals of not more than 0.1 reciprocal nanometers in angle space, and proved adequate to model the spatial coherence effect at 100keV and C_s = 1.8mm (O'Keefe and Sanders, 1975). In the general case, it is possible to automate the process of choosing the sampling interval used by the program to simulate the effects of both spatial and temporal coherence. The objective lens phase change for a defocus of ε and a spherical aberration coefficient of C_s may be written as

$$\chi = \pi\varepsilon\lambda\, k^2 + \pi C_s\, \lambda^3\, k^4/2 \qquad (4)$$

Then the rate of change of χ with k is given by

$$d\chi/dk = 2\pi\, (\varepsilon\lambda k + C_s\, \lambda^3 k^3)$$

and a small change in k produces a small change in χ of

$$\delta\chi = [\, 2\pi\lambda\, (\varepsilon k + C_s\lambda^2 k^3)\,]\, \delta k$$

We wish to choose a value of δk such that $\delta\chi$ will always change by less than a chosen value, say $2\pi/n$. Then

$$\delta k = [n\lambda k (\varepsilon + C_s\lambda^2 k^2)]^{-1} \qquad (5)$$

For any underfocus condition, the plot of χ as a function of k will have a minimum at $\varepsilon = -C_s\lambda^2 k^2$, and a point of inflection at $\varepsilon = -3C_s\lambda^2 k^2$. Depending upon the value of defocus, the maximum rate of change of χ with change in k will occur at either the inflection point or at the largest k value admitted by the objective aperture. Since we desire the largest change in χ to be no more than $2\pi/n$ over any sampling interval in k, the point of maximum rate of change of χ will determine the largest usable sampling step in k. Thus the imaging program should check to find the smaller of the allowed value of δk at the value of k corresponding to the objective aperture radius (equation 5), and the allowed value of δk at the inflection point. This latter value is given by substituting k in equation 5 with an expression for k at the inflection point, to yield

$$\delta k = (-27C_s/\varepsilon^3)^{1/2}/2n \qquad (6)$$

If we choose a value of n of 12, giving a maximum change in χ of 30° between sampling points, then the value of δk at the inflection point is

$$\delta k = (-3C_s/\varepsilon^3)^{1/2}/8$$

For the conditions of Figure 3, the value of δk at the point of inflection is 0.41 reciprocal nanometers, and the value of δk at the objective aperture edge is 0.05 reciprocal nanometers, leading the procedures described above to choose a sample interval of 0.05 reciprocal nanometers. In Figure 3b a δk value of 0.1 reciprocal nanometers was used to produce a satisfactory simulation of the effects of partial spatial coherence, so that the above values for δk are certainly conservative enough for sufficient sampling, and could perhaps be safely doubled to reach the value used in Figure 3b. If we use a radial grid (as in Figure 3b) for sampling the cone of convergence, we find that we need to compute images at 121 or even 169 sampling points for a δk value of 0.05 reciprocal nanometers sampling a convergence cone of semi-angle 1.4 milliradians. On the other hand, modern microscopes have smaller convergence semi-angles, of 0.6 to 1.0 milliradians, and require only 49 or 81 sampling points for adequate simulation.

In addition to partial spatial coherence, image simulations need to include the effects of partial temporal coherence producing a spread-of-focus due both to energy spread in the electron beam and fluctuations in the high-voltage and lens-current power supplies (Fejes, 1977). From equation 4, the desired sampling interval in defocus is

$$\delta\varepsilon = 2(n\lambda k^2)^{-1} \qquad (7)$$

where n is the fraction of 2π allowed as the largest phase change in χ. Using a value for n of 12 in the microscope example used above (100keV microscope with $C_s = 1.8$mm and an aperture at 3.08 reciprocal nanometers), the sampling interval given by equation 7 is 4.7 nm. With a spread-of-focus halfwidth of 8 nm, we then require 9 sampling points in order to extend sampling to two halfwidths on both sides of the spread-of-focus Gaussian.

The example above would then require 121 times 9, or 1089 FFTs to simulate the effects of partial coherence. A modern microscope (400keV with $C_s = 1.2$mm and an aperture at 6 reciprocal nanometers) would require 539 sampling points (a δk of 0.14 reciprocal nanometers at the aperture edge combined with a convergence semi-

angle of 0.6 milliradian produces a requirement for 49 points, whereas a $\delta\epsilon$ of 2.8 nanometers combined with a defocus halfwidth of 7 nanometers requires 11 points).

The size (and hence the time required) of the FFT to be used for the image calculation depends on the number of diffracted beams passing through the objective aperture. For beams out to $\pm(h,k)$, the FFT array size must be $(4h+1)\times(4k+1)$ in order to allow for the doubled frequencies generated when the image amplitude is squared to obtain the image intensity (Saxton, private communication). For a circular aperture, a 16x16 FFT will include up to 37 beams out to $h,k = \pm(3,3)$, a 32x32 FFT will include up to 175 beams out to $\pm(7,7)$, and so on (table 2). For a 20 MFLOPS array processor, a 32x32 FFT typically takes around 4 milliseconds (table 2), so that an image calculation with 539 sampling points and up to 175 beams through the aperture typically requires just over 2 seconds for the FFTs, and about 3 seconds overall, compared with 20 seconds for a reciprocal-space double-sum calculation carried out on a uVAX minicomputer. Even with 751 beams this computation requires less than 10 seconds.

If, instead of a "top-hat" (or a modified "top-hat") weighting, a Gaussian weighting is applied to the spatial coherence sampling, as well as to the spread-of-focus sampling, then the number of FFTs can be reduced by discarding points with double-small weighting, such as the samplings that are both at the edge of the spread-of-focus Gaussian and also in the outer ring of the convergence Gaussian. The program can evaluate the combined weighting and then skip the FFT if the value falls below a selected cutoff.

A further advantage of performing the spatial coherence calculation in real space is that it is possible to include the incident beam convergence not only in the image-forming part of the simulation, but also in the diffraction part. This factor is important in simulating images from thicker crystals.

A Dedicated Image-Simulation System

Typical hardware for running interactive image simulations exists at only a few laboratories. As an example we will describe the facility for electron microscope image analysis and interpretation that has been established at the National Center for Electron Microscopy (NCEM), as a support for the electron microscopes (like the microscopes at the NCEM, the computer facility is available to the user community).

The main component of the facility is a computer system designed for simulation and processing of electron microscope images and we include a description of the system as an example of a typical installation. The host computer is a microVAXII minicomputer running the VMS operating system (figure 5). We chose VMS because this choice gave us the opportunity to develop software of LBL mainframe VAXes while planning and installing the NCEM facility; in addition, the VMS system is multi-user and has virtual memory capability, and it is the operating system most familiar to outside users of the facility. The microVAX is planned to have a 16MB memory (currently it is configured with 13MB), and is reasonably powerful at one MIPS (million instructions per second); it comes with a 71MB Winchester disk (RD53), 95MB tape cartridge backup (TK50), and 5 1/4" floppy disk (RX50). Experience with the image processing facility at Cambridge (Saxton and Koch, 1982) has shown the necessity for a large fast Winchester disk for rapid access to programs and images, with 9-track tape backup for archiving images and reading microdensitometer output. Accordingly, we have added a 858MB disk drive with a formatted capacity of 700 MB and a transfer rate of 3MB/sec (CDC9772), and a tape drive with tri-density capability (Kennedy 9400). The archiving function of the tape drive could be handled by optical disk storage, but the tape unit serves this function

well enough and has the additional advantage of being able to handle input of experimental images digitized on external microdensitometers.

Both simulated and processed images are displayed with a dual-user Gould IP9527. This unit was chosen because of its flexibility. It is configured with four image memory cards, each of 2048x2048x8 bits, and can display images of 1024x1024x8 bits (256 grey levels) at each monitor. Thus each user can store eight 1MB greyscale images (or 32 512x512 images) and have them instantly available for display or manipulation. Additionally, a single user can make use of all the memory to manipulate images up to 4096x4096 pixels (picture elements), and to roam a 1024x1024 display window over the image under trackball control. Alternatively, a single user can work with a true-color image up to 2048x2048 pixels. If it becomes necessary to provide both users with these latter capabilities the memory can be expanded from its present 16MB up to 32 MB. In addition to its display function, the IP9527 has the ability to process images (or parts of images selected by a region of interest) using a DVP (digital video processor) and to compute their intensity histograms.

Because image simulation and image processing require vector operations (including fast Fourier transformation) on large arrays of numbers, an array processor is mandated. In the NCEM system, these functions are carried out by a CSPI MiniMap+4 array processor (AP), consisting of four sub-processors, each with a processing speed of 7MFLOPS (Million Floating Point Operations Per Second). The sub-processors are configured so that a single user can run all four in parallel gaining an 80-fold increase in speed over the microVAX processing speed, or two users can each run two sub-processors, or four users, one each. This AP is thus particularly suitable for a multi-user facility. In addition the CSPI can be configured with up to 48MB of on-board memory; we have 20MB, enough to compute a parameter-20 (1024x1024 point) multislice calculation without host intervention (Table 1), using a multislice algorithm which splits the FFTs among the four sub-processors. Hard copy output of images is by an Apple LaserWriter using the PostScript greyscale routine; examples of typical output can be seen in the NCEMSS images of Figs. 6 and 19, the diffraction pattern of figure 7 and the atomic models of figures 12, 13, and 14.

Improvements in the User Interface

The interface between the program and the user determines to what degree programs are accessible to others besides the author, and how quickly new problems can be tackled. There are numerous examples of powerful software packages that are never used to their potential because of the time it takes the user to become familiar with commands and options. It is also essential that some form of online help be available, with backup available via manuals.

The improvement in speed of the number-crunching aspects of image simulation programs has made it worthwhile to develop an interactive interface with the user; the development of frame-buffer displays and graphics terminals has made it feasible. Ideally, input of data to control the simulation programs is made by selecting menu items displayed on a graphics screen, whereas output of image data is via a frame-buffer monitor for display with subsequent dispatch to a hard copy device. Of course, it is possible to combine the input and output displays when only one display device is available; if windowing software can be implemented there is no loss of flexibility since there are then two "separate" screens, otherwise means must be provided for toggling between the menu (input) screen and the image display (output) screen.

NCEMSS Structure

As an example of a real time, interactive, image simulation program, we can consider NCEMSS (National Center for Electron Microscopy Simulation System), a computer program written to be extremely easy to use by outside users of the

National Center for Electron Microscopy.   It is designed to be fully interactive and to run as close as possible to "real-time" by using a graphics screen for input, an array processor for number-crunching, and a frame-buffer for display of images and diffraction patterns.   Since it is interactive, great flexibility can be built in, allowing the user to change any parameter, toggle the "RUN" button, and allow the program to decide at which stage of the simulation process to pickup the calculation.   In addition, the program will examine the model structure to be simulated, and decide on what size and shape FFT array to use; it will also check the slice thickness and determine the proper number of sub-slices of potential to use in sub-slicing the unit cell.

The choice of a common interface depends on who ultimately will be using the software and there is no single "best" answer.   The most common interface is the command language interpreter (CLI) where the program executes functions according to commands typed at the keyboard.   Parameters associated with a given command are provided with default values that need only be specified if different values are to be used.   Commands are thus quickly entered and executed.   The advantage to this approach is speed, since the user's fingers never need to leave the keyboard.   The disadvantage is the time it takes for new users to learn the commands and options to take full advantage of the software.

A different approach is to provide input through the keyboard using menus. Users respond to prompts by entering their choice of command.   The advantage is that the user does not need to remember any of the commands; the disadvantage is that updating menus takes time and the user needs to be prompted for any parameters.

The interface chosen for NCEMSS makes extensive use of menus, but instead of typing a response from the keyboard, the user responds by moving the cursor to a region associated with the command and presses one of the buttons on a mouse.   The keyboard is used for input of data.   Again the advantage is that the time required to learn to run the software is reduced to almost zero and the graphics screen used with the menus can   be used for drawing curves, crystalline unit cells, etc.   The disadvantage is that experienced users might find it easier and faster to input data from the keyboard.

Another problem which is not related to how commands are input, is determining what information to display and what kind of commands to build into the program.   Generally, new users always find that they require features that are not included, and thus the software is always changing, but with menu input, additional boxes (or even whole new menu pages) are easily added to control new features.   NCEMSS is written such that every menu option has a subroutine associated with it.   This means that only subroutines need to be modified when algorithms change.   Likewise, device-dependent code is limited to a set of primitives, and only simple codes, like those required to draw a line between two points, to read the cursor position, e.g., need to be modified in the case of a change in the display hardware. The penalty in overhead associated with transfer in and out of subroutines is offset by the ease of maintenance.

Several features of NCEMSS are designed to reduce overtime spent consulting manuals and reference books.   The use of menus eliminates the need to remember commands since all options are displayed in plain view.   To obtain help on a specific common, the user need only point at the command and press "h" on the keyboard to produce a help message.   Input of model structures is highly simplified by the fact that scattering factors for the first 98 elements and crystallographic data on all 230 space groups are stored in the program, and the user is prompted for any other necessary information.   The program automatically creates the proper input files such that the user is not required to learn how to edit files.   All parameters needed in a calculation are visible to the user and can immediately be changed.   To ensure that

the program has the correct data there are several ways of drawing the unit cell with the atomic content. Images, diffraction patters, "optical" diffractograms, Pendellösung plots, contrast transfer function plots, etc. are easily calculated and promptly displayed. Images can be zoomed in or out, histogram equalized, Fourier transformed, intensity transformed, filtered and compared, again all with the use of the mouse.

For defect model building, the programs will accept any user-supplied set of symmetry operators. In addition there are menu driven operations that will expand an existing unit cell in any particular direction to create a large supercell and to create defects such as interstitials or vacancies. Atoms can be "picked up" with the mouse, moved around in the unit cell (with x and y coordinates displayed and continuously updated as the mouse is moved) and deposited anywhere. Further facilities for creating defects, such as shearing of subsections, twinning about a given plane etc., will be added in the near future. Figure 6 shows images from a 24.3nm by 0.4nm defect cell containing two interfaces calculated using an array size of 2048x32.

As well as model building and display of simulated images, NCEMSS can display simulated diffraction patterns, index them and show the result of modifying the convergence angle and the camera constant. Figure 7 shows results from Si_3N_4 illustrating the differences between the diffraction pattern and the power spectrum for the case of both untilted and tilted crystals.

It is impossible to present here all the available input screens and output displays of NCEMSS. A sub-set of these is presented in figures 8 through 19. Each figure is largely self-explanatory, so only a brief description is given in the text.

Using NCEMSS

NCEMSS is run at a graphics terminal with an attached local mouse controlling a screen cursor. When the user enters NCEMSS the program writes a File-List menu to the graphics plan of the user's terminal. The File-List menu (Fig. 8) displays a list of the user's personal model-structure (atom position) files with the "PERSONAL" box highlighted. On multiuser systems, the user may obtain a list of model-structure files available in a common file directory by using the mouse to move the screen cursor to the "COMMON" box, and then pressing the mouse button; this action removes the highlight from the "PERSONAL" box, highlights the "COMMON" box, clears the list of personnel file-names and writes a list of common file-names. If the file list extends over more than one page, other pages can be viewed by moving the cursor to the "NEXT PAGE" and "PREVIOUS PAGE" boxes and clicking the mouse button to activate the appropriate box. On A VAX computer running VMS the user can spawn a DCL process from most menus by activating the "DCL" box in the top right of the menu.

To select the desired file, the user activates the "USE DISPLAYED FILE" box, then moves the cursor to the appropriate filename and clicks the mouse. If a new file is desired, then the user activates the "CREATE NEW FILE BOX."

Selecting a filename from the File-List menu produces the File-Parameter menu (Fig. 9) which displays general information about the image-simulation to be computed. If the "CREATE NEW FILE" box of the File-List menu (Fig. 8) had been activated, than an empty File-Parameter menu would be displayed and the user would be prompted to enter the parameters from the keyboard. To change the value of any parameter, the user activates the "CHANGE" box, moves the cursor to the parameter, clicks the mouse, and enters the value from the keyboard; the "CHANGE" box is then de-activated by clicking the mouse on it. Any number of parameters may be changed while the "CHANGE" box is highlighted (activated).

Note that in several cases some parameters are not entirely independent. The number of slices per unit cell is determined by the program from the unit cell parameters, the zone-axis information and the maximum scattering angle to be used (defined by GMAX). If the repeat distance in the electron beam direction is large, the program will use more than one slice per unit cell in order to include the effects of upper Laue layer interactions. If the user wishes to suppress these interactions by setting the number of slices per cell to one, then the program will reset GMAX to an appropriate value.

The parameters appropriate to a number of electron microscopes are stored in a small data file accessed by the program. When the microscope name is changed, these default values are assigned to the parameters CS, DEL, TH and VOLT and displayed. Of course, any of these parameters may be changed individually in the normal way. If the user wishes, he/she can define any microscope name with appropriate associated parameter values, and store this data for recall when the microscope name is entered.

Space groups are selected by the number listed in the International Tables for Crystallography. If no listed group is desired, an entry of zero causes the program to prompt the user to enter any desired symmetry operators from the keyboard. A space group with 2 different specifications, e.g. number 148, can be entered as 148 or 148-1 to obtain the first specification, and 148-2 to obtain the second.

Once all parameters have been set to satisfactory values, the simulation is carried out by clicking the mouse on the "RUN" box. The program indicates the status of the simulation, computes the structure factors out to twice GMAX, forms the projected potential and then the phase-grating to twice GMAX. The "MSLICE" box is then highlighted while the program performs a multislice calculation to the desired foil thickness, using all diffracted beams out to GMAX and storing the complex diffracted beam amplitudes for the selected beam indices. The "IMAGE" box is highlighted while the program selects those diffracted beams falling within the objective aperture radius, applies the phase changes due to C_s and defocus, applies blurring effects due to spread-of-focus (DEL) and incident beam convergence (TH), and writes the image intensity to an image file.

A number of images can be stored in one file by selecting a range of defocus and/or thickness values. A through-focus series of images may be computed by specifying the upper and lower values of defocus and the step size in Angstrom units; e.g. a defocus entry of -400(-50)-700 would produce an image file containing seven images, calculated at values of defocus from -40nm to -70nm in steps of -5nm. Similarly, a foil thickness entry of 200(100)400 would produce images at thickness values of 20nm, 30nm, and 40nm. In combination with the defocus specification above, this thickness specification would result in a file containing 21 images.

If changes have been made only in parameters affecting the electron wavefield emerging from the specimen, (i.e. in parameters like defocus, C_s, or aperture radius, and not in parameters like voltage or Laue circle center), then the PHSGRT and MSLICE steps need not be repeated, and images can be produced by activating the "IMAGE" box, or the "RUN" box. In the latter case, the program will check to see what changes have been made and skip immediately to the "IMAGE" part of the simulation. Similarly, since changes in foil thickness or Laue circle center do not affect the PHSGRT result, activating the "RUN" box will cause the program to start at the "MSLICE" step when "RUN" is activated after these changes are made. Of course, changes in the model structure require a complete new simulation, and activating the "RUN" box after such changes will cause the program to compute all three steps.

Detailed outputs from each step of the simulation are written into files that can be examined by activating the "VIEW FILE" box. In some cases the user may wish to execute only one particular step of the simulation rather than a full simulation. Activating any one of the "PHSGRT", "MSLICE" and "IMAGE" boxes runs just that step of the simulation, and the "VIEW FILE" option may be used to examine the result.

A linear-image contrast transfer function for the current lens conditions may be obtained by activating the "CTF" box. The CTF plot (Fig. 10) is written to the graphics plane of the user's terminal. It shows the undamped CTF for the current values of voltage, defocus and C_s, as well as the damping envelope due to spread-of-focus and incident beam convergence, and also the operating CTF resulting from all the parameters.

A list of all the atoms in the structure model is obtained by activating the "SHOW ATOMS" box. This list (Fig. 11) has menu boxes for paging through the list, as well as boxes for changing parameter values, adding and deleting atoms (by adding and deleting any selected line), and options for building supercells, and for drawing a display of the structure model.

Using the "DISPLAY" box from within the Atom-List menu (Fig. 11), the user can produce atomic-model-like displays of the model structure on the image processors. These displays may be of several unit cells viewed in the electron beam direction (Fig. 12), or of the unit cell of the model viewed in any desire direction, and with the unit cells marked (Fig. 13).

The "BUILD CELL" box in the Atom-List menu (Fig. 11) produces the Build-Structure menu. This menu consists of a display of the unit cell on the graphics terminal (Fig. 14) together with a column of boxes which enable the user to modify the structure by inserting, removing and moving atoms, or changing an atomic species within the original unit cell, or within an incremented unit cell composed of a number of the original cells. Within the display, movements of the cursor within the unit cell (original or augmented) activates a counter showing the cursor position as fractional coordinates of the cell axes. Any changes made using the Build-Structure menu are written to the atom list when the menu is exited, and appear when the Atom-List menu (Fig. 11) is returned to.

Just as the "SHOW ATOMS" box of the File-Parameter menu (Fig. 9) can be used to produce the Atom-List menu (Fig. 11), the "SHOW BASIS" box can be used to produce the Basis-List menu (not shown) to list and change the atoms in the unit cell basis or asymmetric unit.

When the diffraction part of the simulation has been computed, plots of the complex diffracted beam amplitudes may be obtained by activating the "AMPLIT" box of the File-Parameter menu (Fig. 9). The program asks whether the intensity or amplitude of each selected beam should be plotted, and whether the phase plot should be included. In addition, the user is asked whether each plot should be on a fresh screen, or superimposed. Figure 15 shows an intensity/phase plot for the $1\bar{1}1$ and $\bar{1}1\bar{1}$ beams from [110] InP to a crystal thickness of 30nm. Note that the phase values do not approach the single-scattering value of $\pi/2$ for real structure factors at zero-crystal thickness, as a consequence of the lack of a center of symmetry for the structure projected in the [110] orientation chosen.

Once the simulation is complete, images in the output image file may be displayed by activating the "DISPLAY" box of the File-Parameter menu (Fig. 9) to produce the Display menu (Fig. 16). At the top of this menu is a set of four boxes that may be selected to direct the image to one of the two stations of the Gould IP9527 image processor, to the Grinnell GMR274 image processor, or to the user's graphics terminal. The left hand column of boxes shows the function to be displayed; the user

may select from the projected crystal potential, the electron wave function at the exit surface of the specimen, the diffraction pattern at the foil thickness selected in the File-Parameter menu, the computed image, or a digitized experimental image.

The center fifteen boxes of the display menu can be selected to perform various transforms on the image, to display its intensity histogram, to set the number of unit cells displayed, or to Fourier transform it (the Fourier transform of the image intensity is used to produce the "optical" diffractogram or power spectrum as shown in Fig. 7b and d). Once an output function is selected, together with appropriate selections from the center operations boxes, the "EXECUTE" box may be activated to display the image (e.g. Fig. 6) or the diffraction pattern and power spectrum (Fig. 7). If the wrong box is selected (boxes are highlighted when selected) the "CANCEL" box can be activated and the correct boxes reselected.

Whereas images displayed on the image processor screens are displayed with a grey scale with 256 levels (the image processors have 8 bit-planes of display memory), images displayed on the graphics terminal (only one bit-plane) required a "dithered" greyscale. Figure 17 shows a dithered image, output with 26 grey levels formed by a 5x5 cell of points that are switched to either black or white. Such images are usually only for quick inspection when the images processors are in use by other users.

One selectable operation in the Display menu (Fig. 16) is to adjust the image contrast. Selection of the "ADJ.CNT" box produces the contrast menu of Fig. 18. The transfer plot shows how image intensities on the horizontal axis are mapped to the output intensities of the vertical axis. For the condition shown, the mapping is 1:1, and the contrast (slope of the transfer line) is defined as having the value 1.0, with a brightness of 0.5 units (center point of the transfer line). The user may decrease or increase brightness with the "DECR" or "INCR" boxes, and change the contrast value with "UP" or "DOWN". Using these boxes changes the corresponding bar-graph, and also redraws the transfer line. Note that dropping the contrast below zero reverses the image contrast to give a negative image. As well as adjusting the contrast and brightness, the user has the further option of drawing the transfer line directly by first activating the "SET BLACK-WHITE" box, then placing the cursor at some point within the transfer line area and clicking the mouse, and again placing the cursor at a second position on the desired transfer line and clicking again. The program then draws a new transfer line through the two selected points (extending it, if necessary, to intersect the plot aces), at the same time adjusting the two bar graphs.

When directed to the Gould IP9527, the resultant images can be arranged on the display screen as one picture consisting of a set of montage images with titles. The contents of the display screen can then be sent to the Apple LaserWriter for output (Fig. 19).

Conclusions

In the last few years, the advent of inexpensive array processors has enabled quite modest host computers to attain close to interactive processing of electron image simulation programs, especially when suitable algorithms are written to use the array processor efficiently. Secondly, the development of graphics terminals has enable the use of software to run the programs interactively. Thirdly, the availability of frame-buffers provides instant display of results, with hard copy output available immediately on a laser printer.

Acknowledgements

Discussion with Reviewers

Reviewer: In describing the FFT multislice, the authors should give due reference to the first published report of this method by Ishizuka and Uyeda (1977).

Authors: The report by Ishizuka and Uyeda (1977) is unsatisfactory in that it incorrectly stresses that the number of diffracted beams must be no less than the number of phase-grating coefficients in order to avoid the effects of aliasing. This confusion appears to have arisen when the concept of FFT multislice was introduced to the Kyoto group by a visiting researcher familiar with its use in the Physics Department of Melbourne University. At Melbourne, it had been found that aliasing occurred when the convolution step of the reciprocal-space multislice (equation 1) was replaced by a forward/inverse transform sequence. This aliasing boosted the amplitudes of the outer beams in the calculation until "knock off" parameters were introduced to set the outer beam amplitudes to zero after each slice. However, even after this procedure eliminated the aliasing effect, results from the FFT multislice did not exactly match ones from the "standard" reciprocal-space formulation (O'Keefe, 1972). For this reason, the SHRLI80 programs (O'Keefe et al, 1978) continued to use the (slower) reciprocal-space form of the multislice. Experiments with various scattering algorithms (Self et al, 1983) led to the testing of different ways of avoiding aliasing in FFT multislice. It was realized that one such method was by choosing a $q(x)$ formed from $Q(k)$ terms that extended only 2/3 of the way to the array edges, provided that only the same number of diffracted beams was used in $\Psi(k)$, with the outer 5/9 of the diffracted beam array reset to zero after every slice. Another method was to ensure that $q(x)$ was formed from a $Q(k)$ that extended out to twice the scattering angle as $\Psi(k)$, and to reset the outer 3/4 diffracted beams to zero after each slice. This latter method conforms to the physics of the scattering process, and proved to yield the same results as the reciprocal-space multislice; a multislice program using this method was incorporated in the SHRLI81 simulation programs.

/

References

Ackland D, Dahmen U, Echer CJ, Kilaas R, Krishnan KM, Nelson C, O'Keefe MA, Smith W, Turner J (1986) Recent TEM Applications. J. Metals 38, 19-24.

Coene W, Van Dyck D (1984) The real space method for dynamical electron diffraction calculations in high resolution electron microscopy. II. Critical analysis of the dependency of the input parameters. Ultramicroscopy 15, 41-50.

Fejes P (1977) Approximations for the calculation of high-resolution electron microscope images of thin films. Acta Cryst. A33, 109-113.

Frank J (1973) The envelope of electron microscope transfer functions for partially coherent illumination. Optik 38, 519-536.

Goodman P, Moodie AF (1974) Numerical evaluation of N-beam wave functions in electron scattering by the multislice method. Acta Cryst. A30, 322-324.

Iijima S, O'Keefe MA (1977) Interpretation of dark field images of crystals. In 35th Ann. Proc. EMSA, G. W. Bailey (ed.), Claitor's, Baton Rouge, LA, 190-191.

Ishizuka K, Uyeda N (1977) A new theoretical and practical approach to the multislice method. Acta Cryst. A33, 740-749.

Kilaas R (1987a) Interactive software for simulation of high resolution TEM images, Proc. 22nd MAS, R. H. Geiss (ed.), Kona, Hawaii, 293-300.

Kilaas R (1987b) Interactive simulation of high resolution electron micrographs. In 45th Ann. Proc. EMSA, G.W. Bailey (ed.), Baltimore, Maryland, 66-69.

Kilaas R, O'Keefe MA, Krishnan KM (1987) On the inclusion of upper Laue layers in computational methods in high resolution transmission electron microscopy. Ultramicroscopy 21, 47-62.

Kilaas R, Gronsky R (1983) Real space image simulation in high resolution electron microscopy. Ultramicroscopy 11, 289–298.

O'Keefe MA (1972) Lattice images of large-unit-cell oxide crystals by n-beam theory. M.Sc thesis, University of Melbourne.

O'Keefe MA (1976) Interpretation of electron microscope crystal lattice images. Ph.D. Thesis, University of Melbourne.

O'Keefe MA (1979) Resolution-damping functions in non-linear images. In 37th Ann. Proc. EMSA, G. W.Bailey (ed.), San Antonio, Texas 556-557.

O'Keefe MA (1984) Electron image simulation: a complementary processing technique. In Electron Optical Systems, J. J. Hren, F. A. Lenz, E. Munro, P. B. Sewell (eds.), SEM Inc., AMF O'Hare,IL 60666, 209-220.

O'Keefe MA, Buseck PR, Iijima S (1978) Computed crystal structure images for high resolution electron microscopy. Nature 274, 322-324.

O'Keefe MA, Iijima S (1978)   Calculation of structure images of crystalline defects, in Electron Microscopy 1978, J. M. Sturgers (ed.), Microscopical Society of Canada, Toronto, 282-283.

O'Keefe MA, Sanders JV (1975) N-beam lattice images. VI. Degradation of image resolution by a combination of incident-beam divergence and spherical aberration. Acta Cryst. $\underline{A31}$, 307-310.

Saxton WO, Koch TL (1982)   Interactive image processing with an off-line minicomputer: organization, performance and applications. J. Microscopy $\underline{127}$, 69-75.

Self PG, O'Keefe MA, Buseck PR, Spargo AEC (1983) Practical computation of amplitudes and phases in electron diffraction.   Ultramicroscopy $\underline{11}$, 35-52.

Van Dyck D (1980)   Fast computational procedures for the simulation of structure images in complex or disordered crystals: A new approach. J. Microscopy $\underline{119}$, 141-152.

Van Dyck D (1983)   High-speed computation techniques for the simulation of high resolution electron micrographs.   J. Microscopy $\underline{132}$, 31-42.

Van Dyck D, Coene W (1984)   The real space method for dynamical electron diffraction calculations in high resolution electron microscopy. I. Principles of the method. Ultramicroscopy $\underline{15}$, 29-40.

Param	Array Size	#Beams	Memory Required (Bytes)		
			Recip.	Real	FFT
14	128 x 128	4K	224K	256K	288K
15	128 x 256	8K	448K	512K	576K
16	256 x 256	16K	896K	1024K	1152K
17	256 x 512	32K	1.75M	2M	2.25M
18	512 x 512	64K	3.5M	4M	4.5M
19	512 x 1024	128K	7M	8M	9M
20	1024 x 1024	256K	14M	16M	18M

Table 1. Array processor memory requirements for in-memory FFT multislice compared with reciprocal-space and real-space multislice algorithms. For a phase-grating array of $2^n \times 2^m$, the multislice size parameter is n+m and the number of beams scattered through the crystal is 2^{n+m-2}. The three complex arrays common to all three algorithms are the phase-grating array ($8 \times 2^{n+m}$ bytes), the diffracted beam (or wave function) array ($8 \times 2^{n+m-2}$ for the reciprocal-space multislice, and $8 \times 2^{n+m}$ for the FFT and real-space multislices), and the propagator array ($8 \times 2^{n+m-2}$ bytes for the reciprocal space and FFT multislices, and thirteen for the real space multislice). In addition, the reciprocal-space multislice algorithm needs to store input and output wave functions separately, and thus requires an extra $8 \times 2^{n+m-2}$ bytes. Additional minor memory requirements not included in the table are the 13 terms of the real-space algorithm's propagator, and the sin/cos tables for the FFT multislice algorithm. Single-precision arithmetic (32 bit real, 64 bit complex) is assumed.

Max h,k	2h+1,2k+1	Array Size	Beams	FFT (sec)
3,3	7,7	16 x 16	37	.001
7,7	15,15	32 x 32	175	.004
15,15	31,31	64 x 64	751	.015
31,31	63,63	128 x 128	3115	.057

Table 2. Index range, array sizes and FFT times for partial coherence calculations in an array processor of 20 MFLOPS processing power.
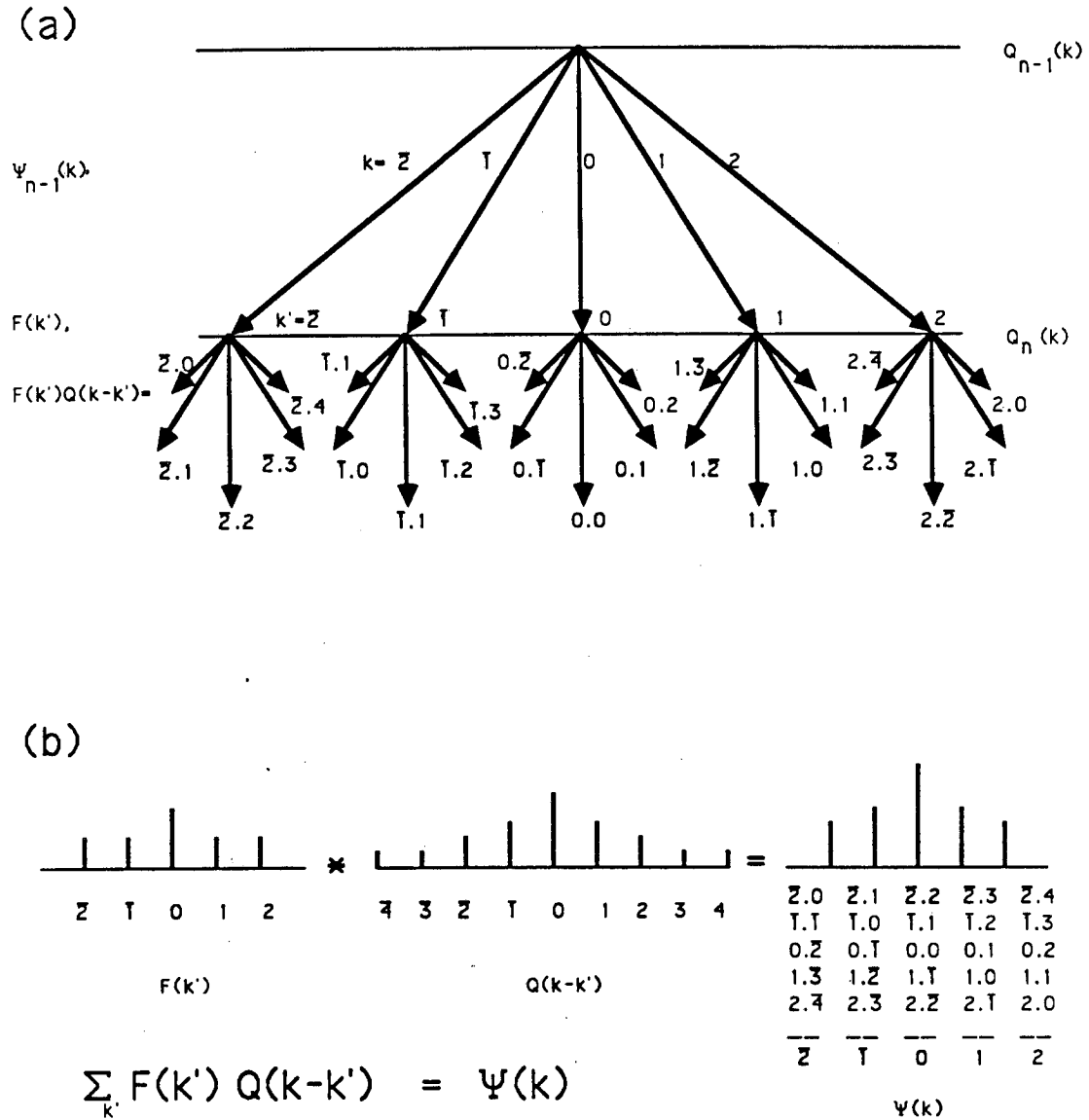
Fig.1. Convolution description of multiple scattering. In order to include all scattering contributions to the dynamic electron wavefield $\Psi(k)$, the phase-grating coefficients $Q(k)$ must extend out twice as far in reciprocal space as do the coefficients of the wavefield. For the five depicted diffracted beams $\Psi(k)$, (where $|k| = \bar{2},\bar{1},0,1,2$), the convolution requires nine phase-grating coefficients $Q(k)$, (where $|k| = \bar{4}$ to 4). (a) sketch of the contributions to the five diffracted beam directions; note that (e.g.) the contribution of the incoming $\Psi(2)$ beam to the outgoing $\Psi(\bar{2})$ beam requires the presence of the $Q(\bar{4})$ phase-grating coefficient. (b) Delta-function representation of the scattering process, showing each $F(k').Q(k-k')$ term contributing to the outgoing $\Psi(k)$, $|k| = \bar{2}$ to 2. Here the incoming function $F(k)$ is equal to the product $\Psi(k).P(k)$ in equation (1). Note the presence of the $Q(\pm3)$ and $Q(\pm4)$ coefficients in the table of interactions; except for $\Psi(0)$, every outgoing $\Psi(k)$ contains a contribution from these outer coefficients, and would be in error without them.
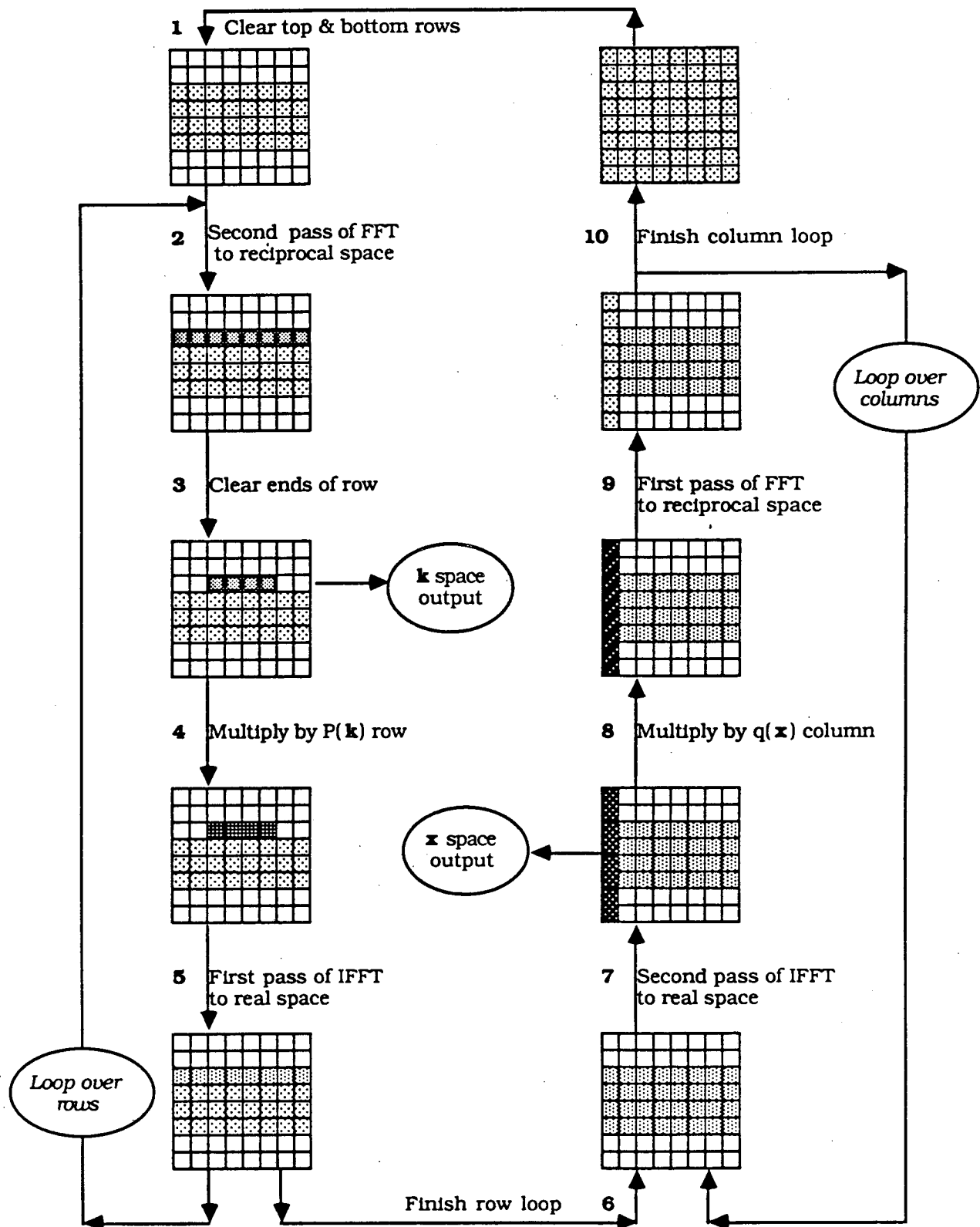
Array Processor FFT Multislice Algorithm

1 Clear top & bottom rows

2 Second pass of FFT to reciprocal space

10 Finish column loop

Loop over columns

3 Clear ends of row

k space output

9 First pass of FFT to reciprocal space

4 Multiply by P(**k**) row

x space output

8 Multiply by q(**x**) column

5 First pass of IFFT to real space

7 Second pass of IFFT to real space

Loop over rows

Finish row loop   **6**

Fig.2. Flow chart for the FFT multislice algorithm using one-dimensional FFTs to reduce the amount of computation compared with any algorithm using two-dimensional FFTs. This algorithm is particularly suitable for use with an array processor since it also minimizes the number of data transfers into and out of the array processor.
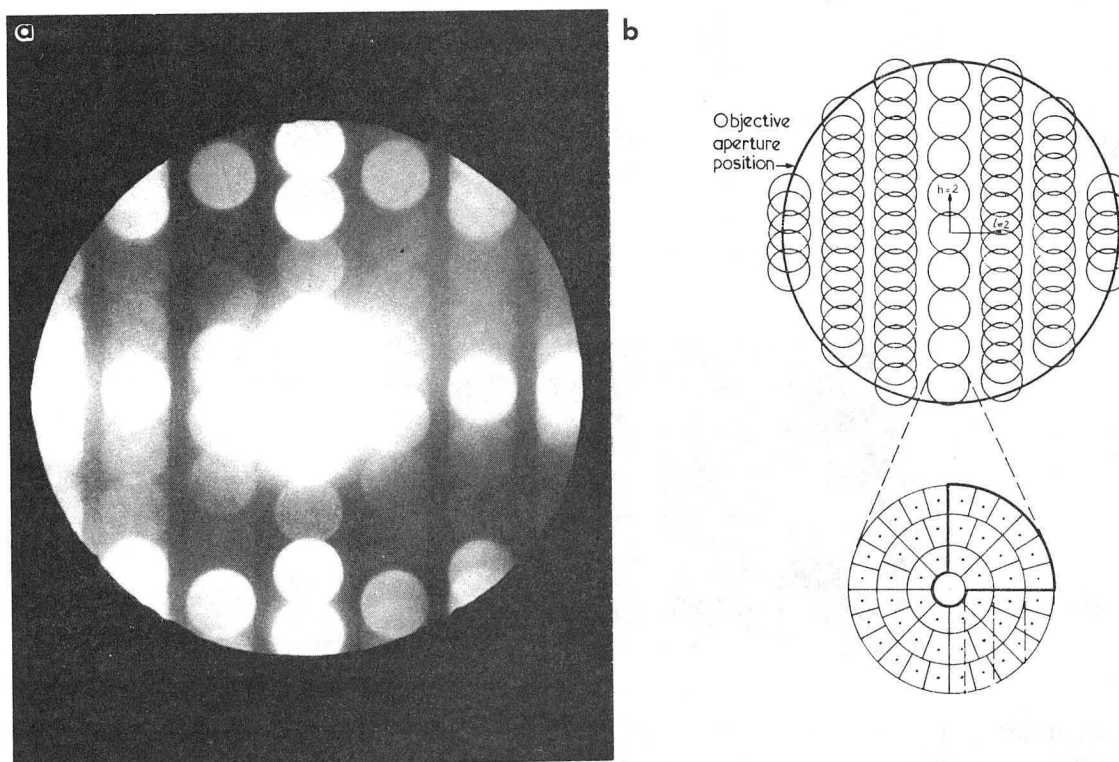
Fig.3. Simulation of beam convergence by real space sampling. (a) 001 diffraction pattern of $Nb_{12}O_{29}$ obtained with illumination adjusted for imaging, and showing discs due to electron beam convergence in imaging mode. (b) Method of including beam convergence in the image simulation. Each disk admitted by the objective aperture is sampled at 49 points and images calculated at angles corresponding to each sampling point; these images are then summed to produce the final overall image. Areas of disks falling outside the objective aperture are excluded from the summation to produce the overall image (O'Keefe and Sanders, 1975).
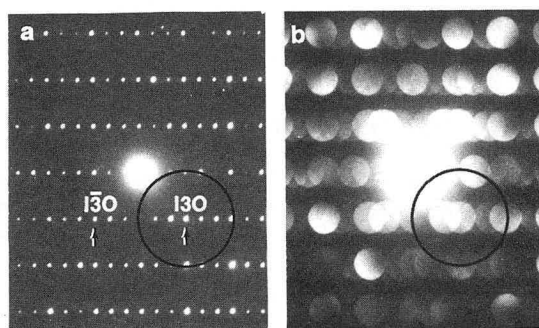
Fig. 4. Diffraction patterns from [001] $4Nb_2O_5.9WO_3$ for parallel (a) and focussed (b) illumination. The objective aperture size and position are shown. Note the many diffraction cones intersected by the aperture under the dark-field imaging conditions (Iijima and O'Keefe, 1977).
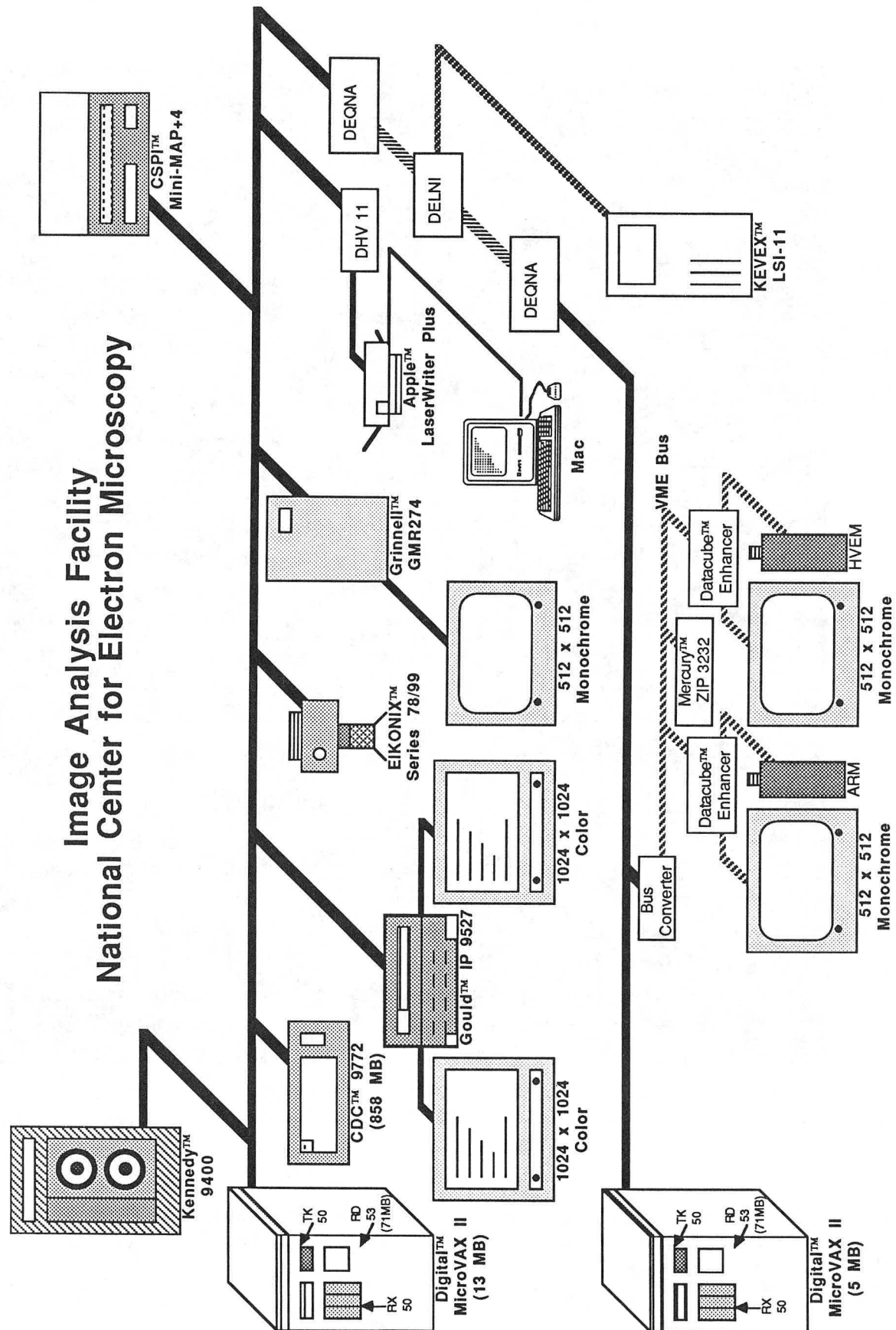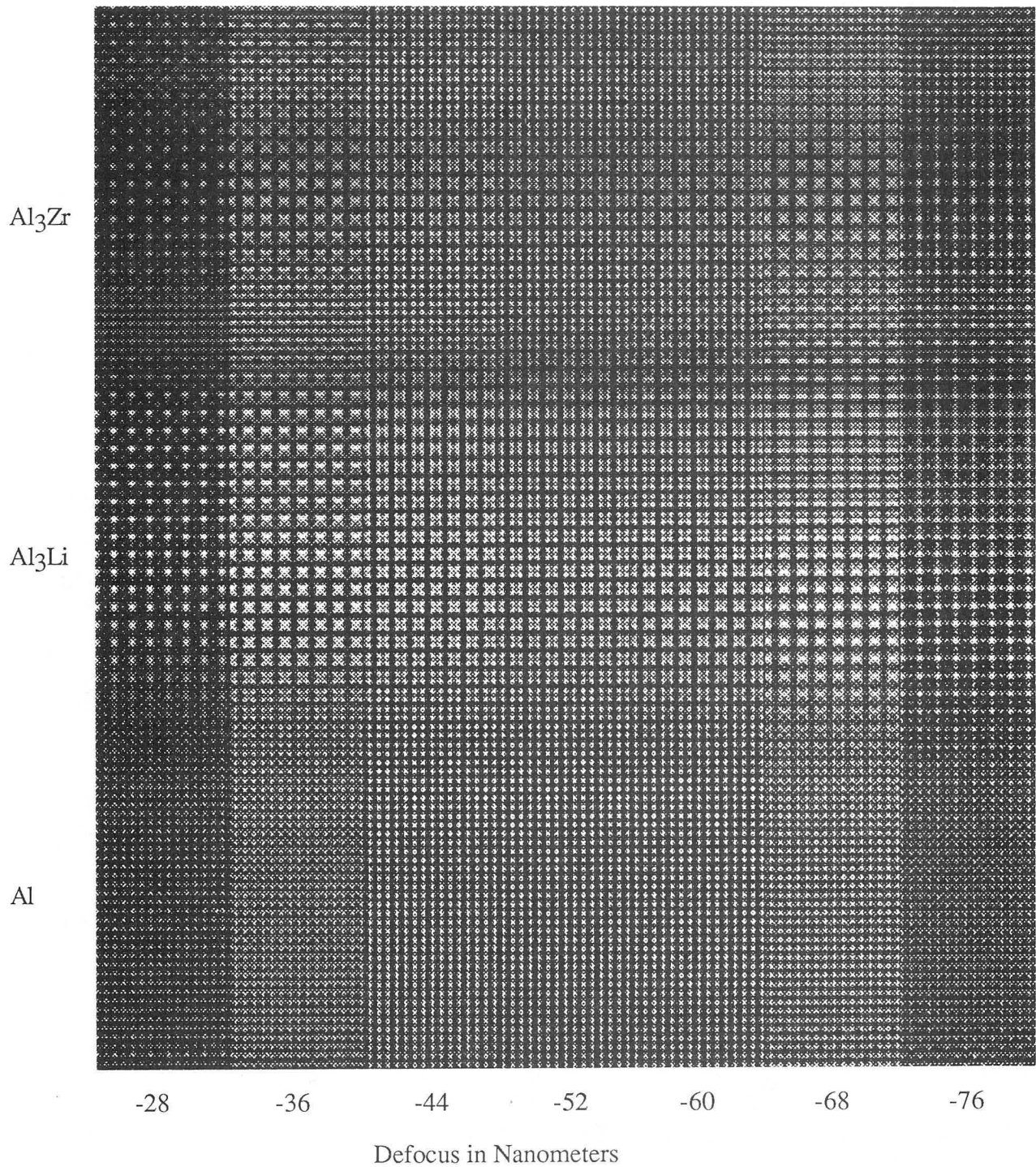
25

Image Analysis Facility
National Center for Electron Microscopy

Fig. 5. Diagram of the NCEM image simulation/image processing computer system.

Al₃Zr

Al₃Li

Al

Defocus in Nanometers

Fig. 6. LaserWriter output of a through-focus series of images (defocus marked in nm) of a $Al_3Zr/Al_3Li/Al$ double-boundary structure. The NCEMSS program chose a 2048x32 array to simulate the 24.3nm by 0.405nm unit cell. The images were simulated for a 120A thick specimen imaged in the NCEM Atomic Resolution Microscope (ARM), using ARM parameters of 1000keV, 2.8mm spherical aberration, 0.6 milliradian semi-angle of beam convergence (corresponding to the 100μm condenser aperture) and 10nm half-width of spread of focus.

27
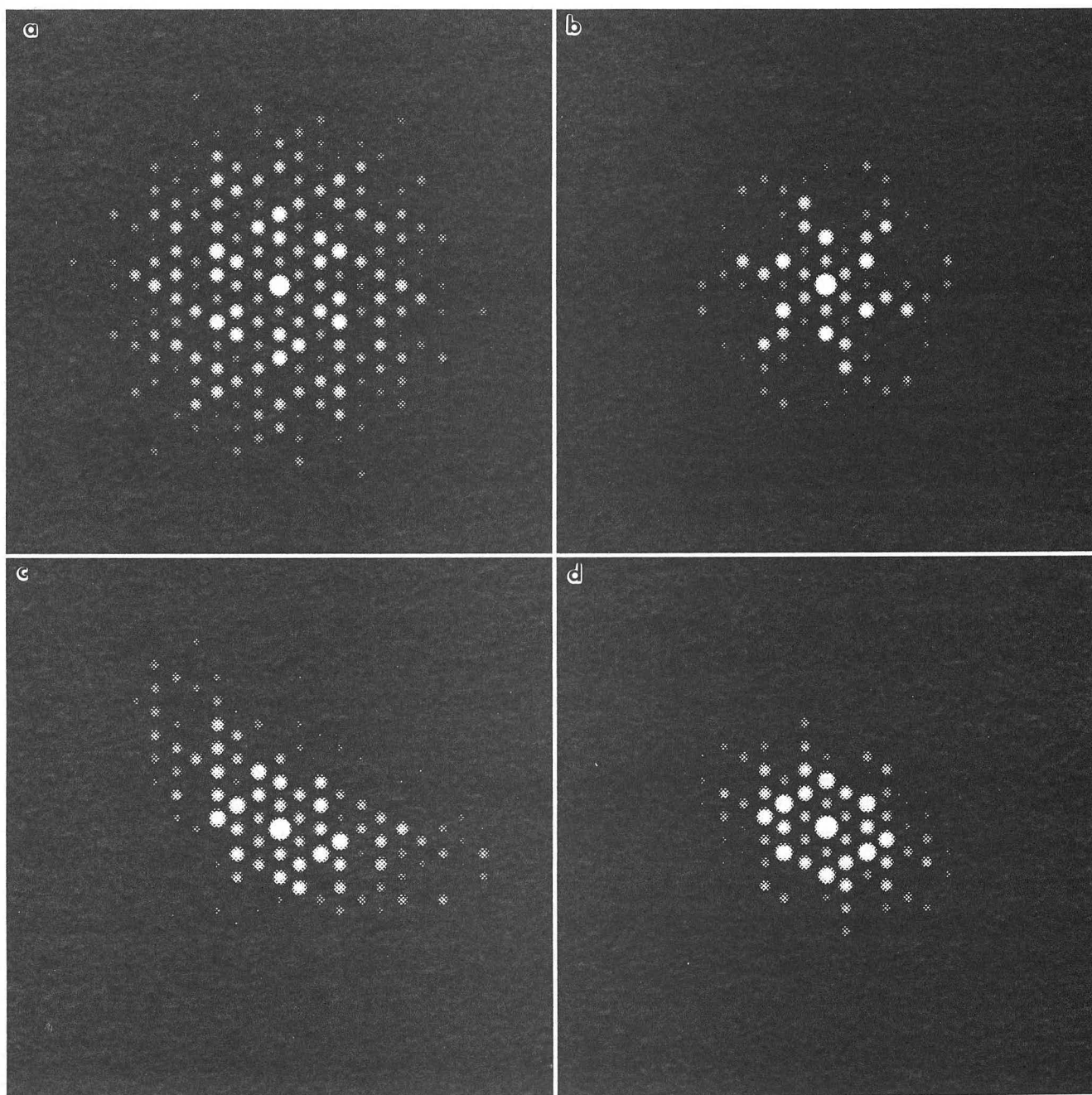
Fig. 7. LaserWriter output of reciprocal space results from NCEMSS. (a) Diffraction pattern from [001] Si_3Ni_4. (b) Power spectrum from image simulated from (a) for ARM conditions. (c) Diffraction pattern from Si_3N_4 tilted 12 milliradian from [001]. (d) Power spectrum from image simulated from (c).

8					
PERSONAL	COMMON	NEXT PAGE	PREV. PAGE	DCL	

AL2O3	AL3LI	AL3LIVAC	ALUMINUM
ANATESE	BATI2O5	BATIO3	BITE
BITE4	BITEL1	BITEL4	BTEK
BTEKNEW	CDTE	CO2SIO4	COPPER
CUBE	CUBE121	CUBELARGE	CULARGE
CUTI2	D100	DOLOM	FAUJ001
FAUJ110	FAUJPT	GAALAS	GAAS
GENBO	GOLD	INP	INTERFACE
INTERFACE1	LASRCUO4	LAYER1	LAYER2
MAGNESIUM	MOLYBDENUM	MZR	MZRO2
OCTAHEDRON	PLATINUM	QW	QWLARGE
RUTILE	SI3N4	SI3N4IFC	SICBETA
SILICON	SIO2	SIO2EXT2	SPINEL
SPINEL4	STACY	TEST	TIO2B
ZIRCON	ZSM5		

USE DISPLAYED FILE	CREATE NEW FILE	RETURN

Fig. 8. NCEMSS File-List menu showing a list of user's atom coordinate or structure-model files. The boxes are used to control program options by means of a mouse-directed cursor.

```
9                                                                    ┌──────┐
                                                                     │ DCL  │
  INPUT FOR : ORTHORHOMBIC ZRO2          SPACE GROUP #  61            └──────┘

  A  =        10.01                      ALPHA  =      90.00

  B  =         5.24                      BETA   =      90.00

  C  =         5.05                      GAMMA  =      90.00

  GMAX =       2.00                      ZONE AXIS :  1 1 0

  NO. OF SYMMETRY OPERATORS    :    8    NO. OF SLICES PER UNIT CELL :      1

  NO. OF ATOMS IN THE BASIS :      3     NO. OF DIFFERENT ATOMS :           2

  MICROSCOPE : ARM                       FOIL THICKNESS :              600.00

                    CS   =     2.80      AMPLIT. OUTPUT FOR PLOTTING :   YES

                    DEL  =   100.00      THE INDICES ARE :      H    K    L

                    TH.  =     0.30                             0    0    0
                                                               0    2    0
  VOLT.  =  1000.00                                            2    0    0

  CENT. OF LAUE CIRCLE: H =      0.00

                       K =      0.00

  OBJ. LENS DEFOCUS :  -1200.00

  APERTURE RADIUS   :      0.70

  CENT. OF OBJ. APRT. : H =      0.00    CENT. OF OPTIC AXIS : H =       0.00

                       K =      0.00                         K =       0.00


  ┌──────────┐  ┌────────────┐  ┌──────────┐  ┌──────────┐  ┌──────────┐  ┌──────────┐
  │  CHANGE  │  │ SHOW BASIS │  │  PHSGRT  │  │  IMAGE   │  │ DISPLAY  │  │   CTF    │
  └──────────┘  └────────────┘  └──────────┘  └──────────┘  └──────────┘  └──────────┘
  ┌──────────┐  ┌────────────┐  ┌──────────┐  ┌──────────┐  ┌──────────┐  ┌──────────┐
  │   RUN    │  │ SHOW ATOMS │  │  MSLICE  │  │  AMPLIT  │  │ VIEW FILE│  │  RETURN  │
  └──────────┘  └────────────┘  └──────────┘  └──────────┘  └──────────┘  └──────────┘
```

Fig. 9. NCEMSS File-Parameter menu, arrived at by selecting a file from the File-List menu. The File-Parameter menu is the main NCEMSS menu; it displays current values of many of the image simulation parameters, can be used to change them, and contains control boxes for running (all or part of) the simulation, generating a CTF plot and calling up other menus.

CONTRAST TRANSFER FUNCTION

Accel.Voltage = 1000.0 kV
Spher. Abber. = 2.8 mm
Obj. Lens Def. = -600 A
Delta = 100 A
Div. Angle [Half] = 0.60 mrad
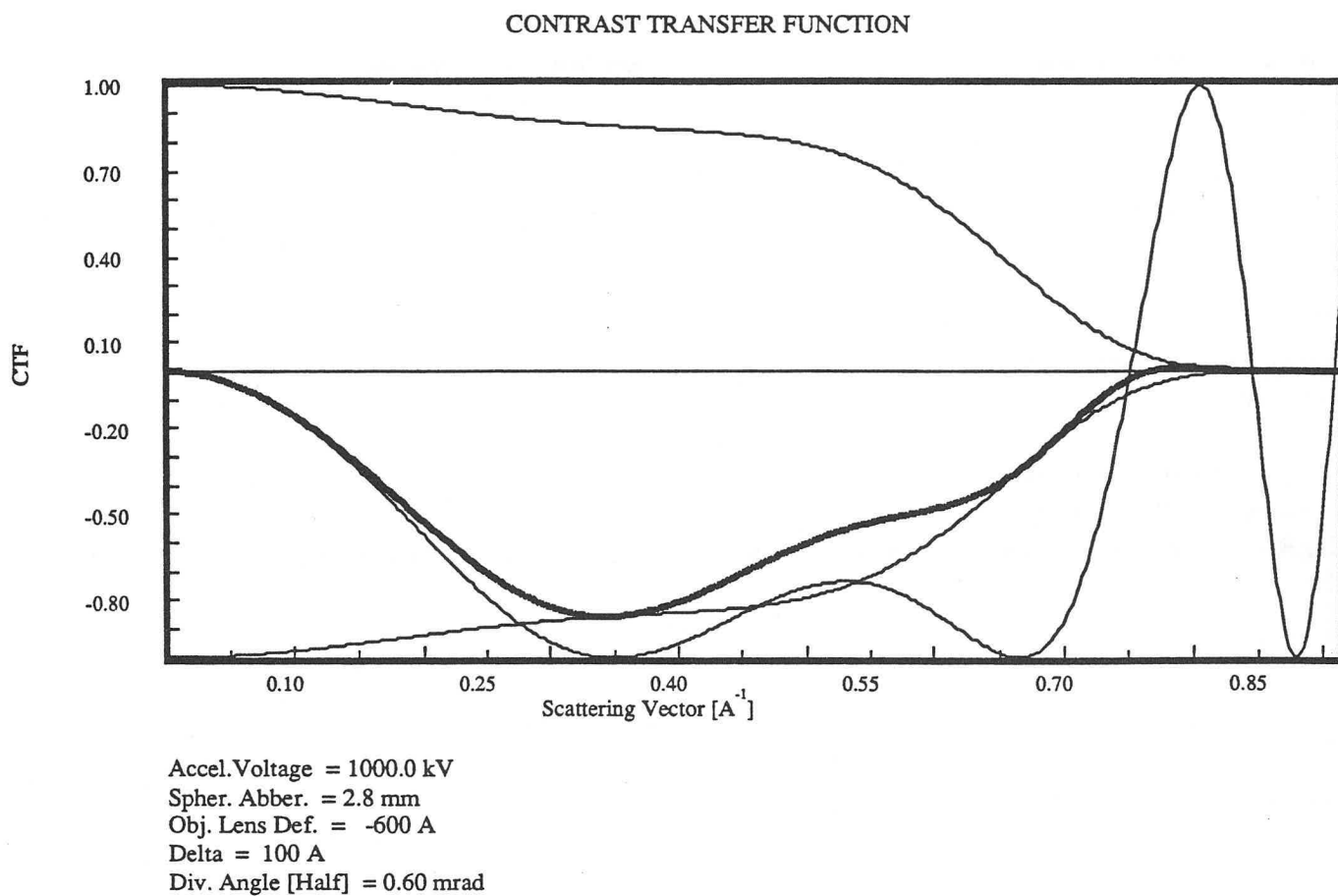
Fig. 10. Linear-image contrast transfer function (CTF) plot, showing undamped CTF, damping function, and operating (damped) CTF for values of parameters set in the File-Parameter menu. This plot is displayed on the user's graphics terminal, and is generated by activating the CTF box of the File-Parameter menu.

#	TYPE	NAME	X	Y	Z	DW	OCC
1	1	ZR	0.1340	0.0334	0.2500	0.2000	1.00
2	1	ZR	0.6340	0.4666	0.7500	0.2000	1.00
3	1	ZR	0.8660	0.5334	0.2500	0.2000	1.00
4	1	ZR	0.3660	0.9666	0.7500	0.2000	1.00
5	1	ZR	0.8660	0.9666	0.7500	0.2000	1.00
6	1	ZR	0.3660	0.5334	0.2500	0.2000	1.00
7	1	ZR	0.1340	0.4666	0.7500	0.2000	1.00
8	1	ZR	0.6340	0.0334	0.2500	0.2000	1.00
9	2	O	0.0411	0.3750	0.1292	0.2000	1.00
10	2	O	0.5411	0.1250	0.8708	0.2000	1.00
11	2	O	0.9589	0.8750	0.3708	0.2000	1.00
12	2	O	0.4589	0.6250	0.6292	0.2000	1.00
13	2	O	0.9589	0.6250	0.8708	0.2000	1.00
14	2	O	0.4589	0.8750	0.1292	0.2000	1.00
15	2	O	0.0411	0.1250	0.6292	0.2000	1.00
16	2	O	0.5411	0.3750	0.3708	0.2000	1.00
17	2	O	0.2280	0.7500	0.5000	0.2000	1.00
18	2	O	0.7280	0.7500	0.5000	0.2000	1.00
19	2	O	0.7720	0.2500	0.0000	0.2000	1.00
20	2	O	0.2720	0.2500	0.0000	0.2000	1.00

PAGE +

PAGE -

PAGE #

CHANGE   ADD   DELETE   BUILD CELL   DISPLAY   RETURN

Fig. 11. NCEMSS Atom-List menu, arrived at by activating the SHOW ATOMS box on the File-List menu. This menu shows the current atom position parameters for the model structure. The example shown lists the first 20 atoms; other pages of the list can be viewed by using the right-hand boxes. The lower boxes can be used to modify the structure.
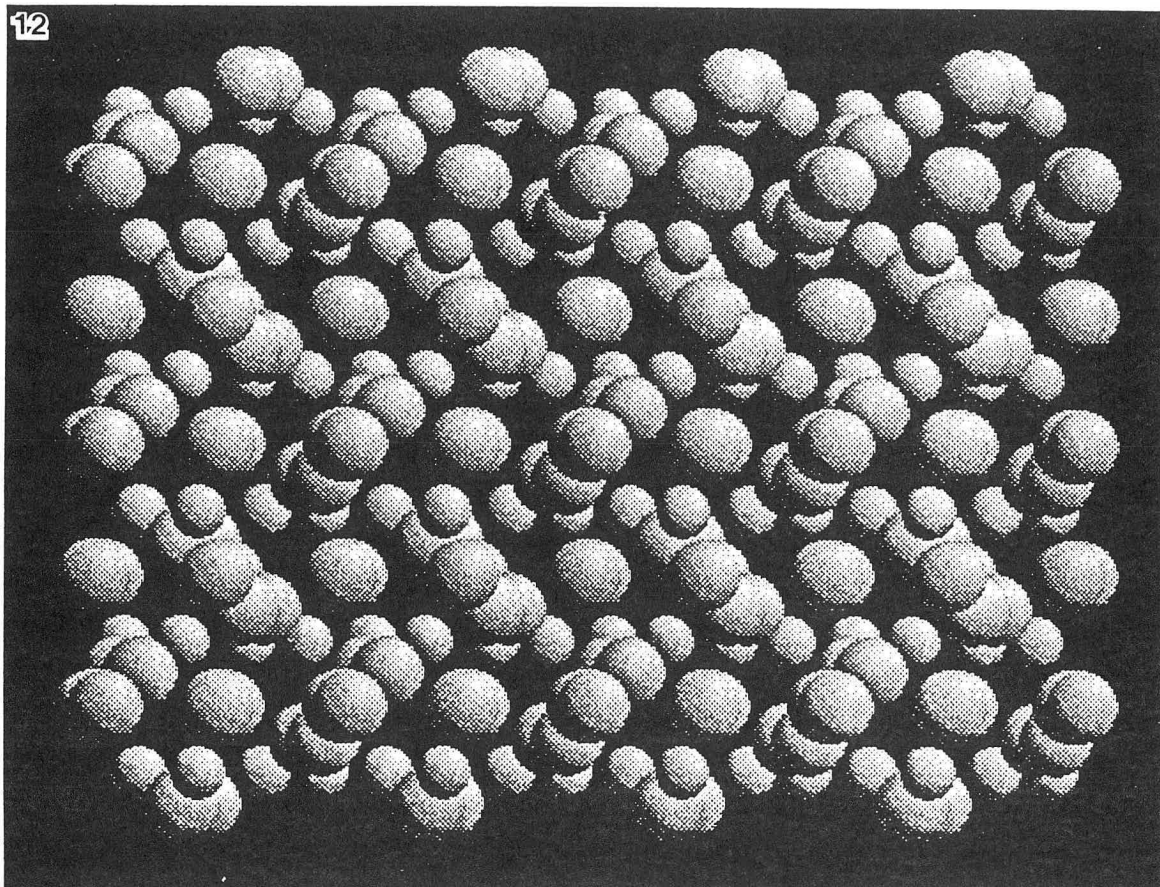
Fig. 12. Multi-cell model structure display on the Gould IP9527 frame buffer, produced by activating the DISPLAY box of the Atom-List menu, and then setting some parameters in response to prompts.
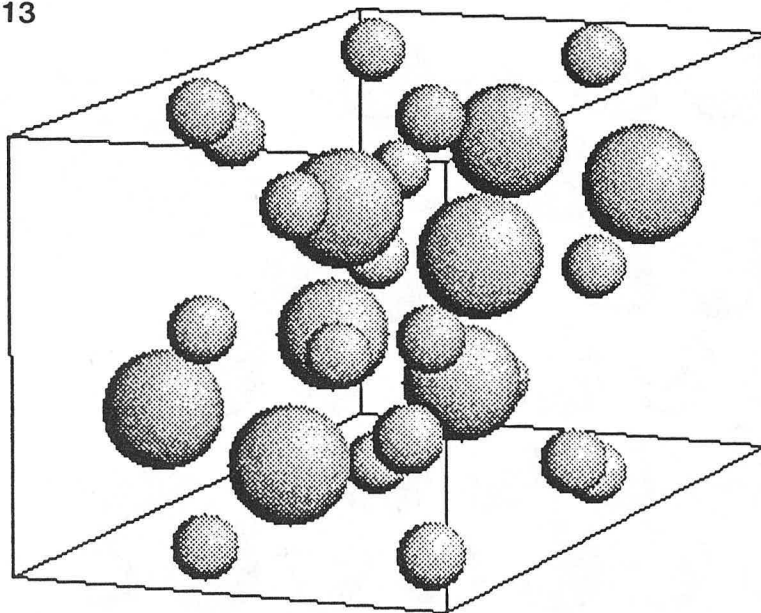
13

Fig. 13. Single unit-cell display on the frame buffer. The display was produced as for Fig. 12, but with different responses to prompts.
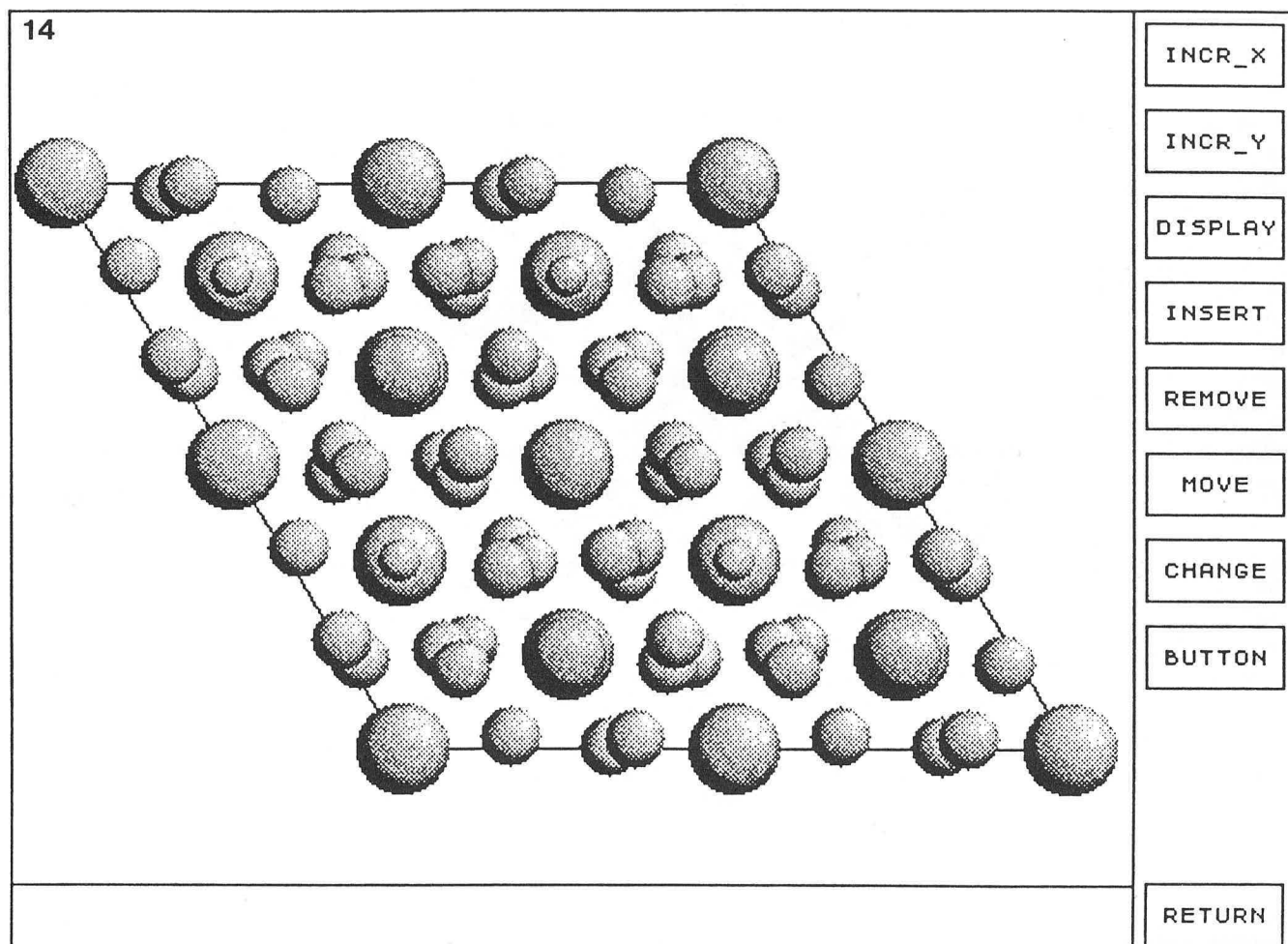
Fig. 14. NCEMSS Build-Structure menu, arrived at by activating the BUILD CELL box on the Atom-List menu. Boxes available can be used to increase the cell size, add and remove any atom, change the atomic species of any atom, or move any atom with a mouse-directed cursor.
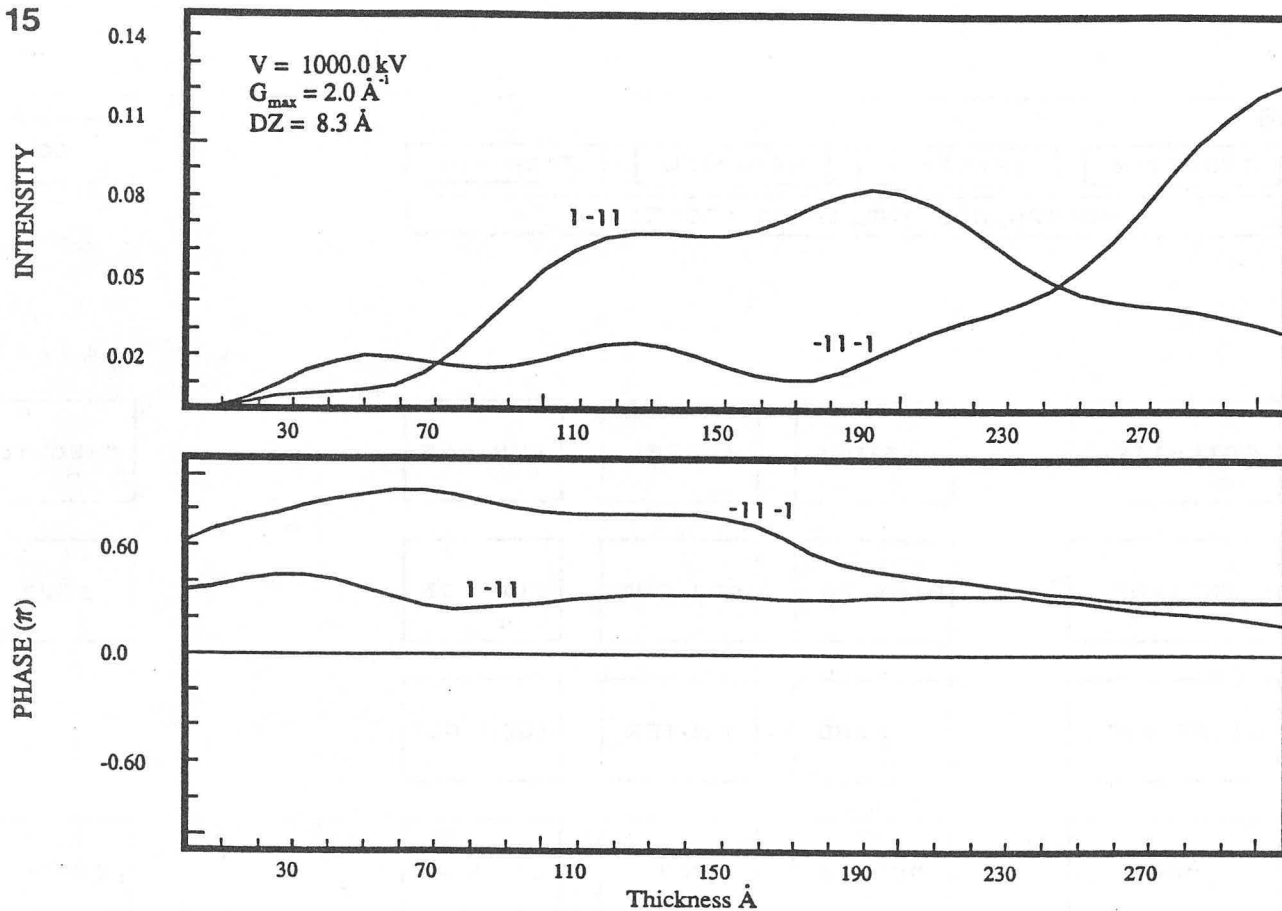
Fig. 15. Plot of the intensities and phases of the $1\bar{1}1$ and $\bar{1}1\bar{1}$ beams from [110] InP at 100keV for a calculation including all beams out to a G_{max} value of 20 reciprocal nanometers (phase-grating coefficients computed out to 40 reciprocal nanometers), and a slice thickness of 0.83nm.

Fig. 16. NCEMSS Image-Display menu, arrived at by activating the DISPLAY box on the Parameter-List menu. This menu is used to choose an input function (right-hand column of boxes), apply an operation (center 15 boxes), and output the result to the selected output device (top row of boxes).
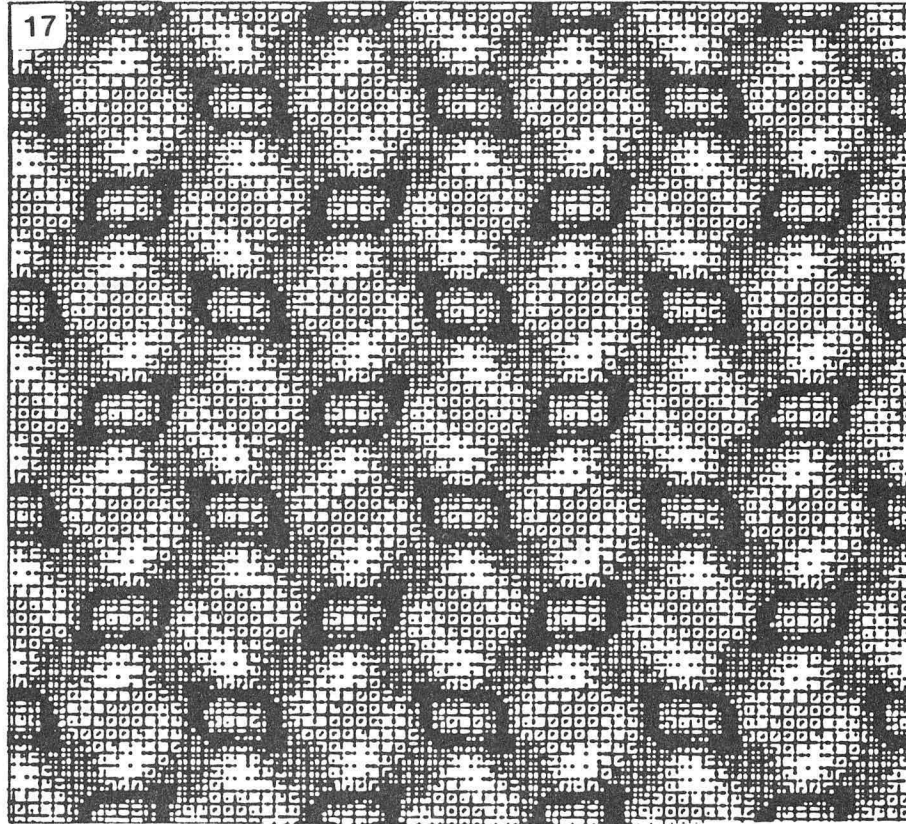
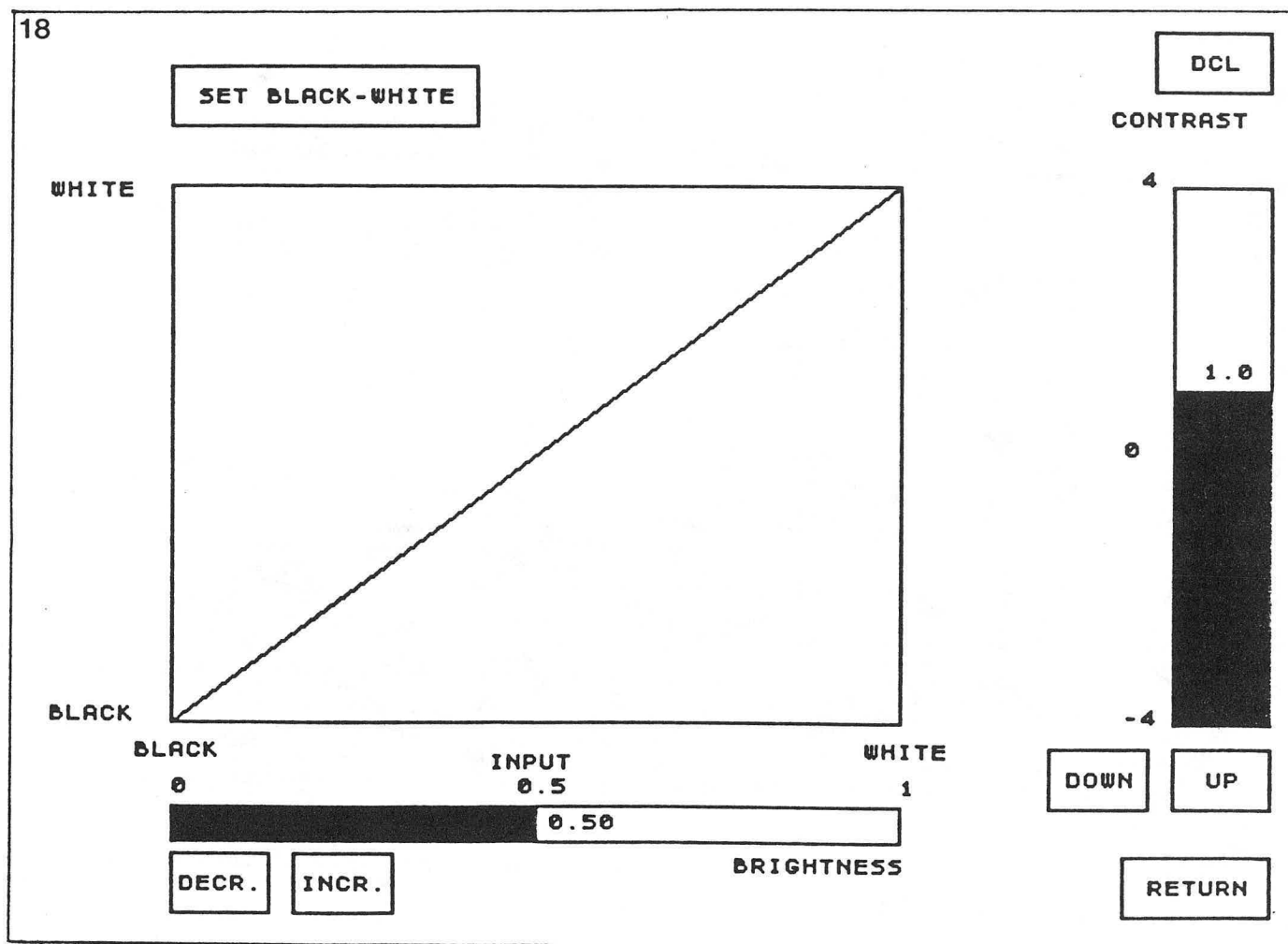**Fig. 17.** A NCEMSS image output on the graphics terminal using a 5x5 element dithered grey scale.

Fig. 18. NCEMSS Set-Contrast menu, arrived at by activating the ADJ.CNT operation on the Image-Display Menu. This menu is used to adjust the output image contrast by adjusting the transfer line used to map incoming image intensities to outgoing image intensities. The plot is shown for default conditions, with a contrast value (slope of line) of one and a brightness value (center point of line) of one-half.
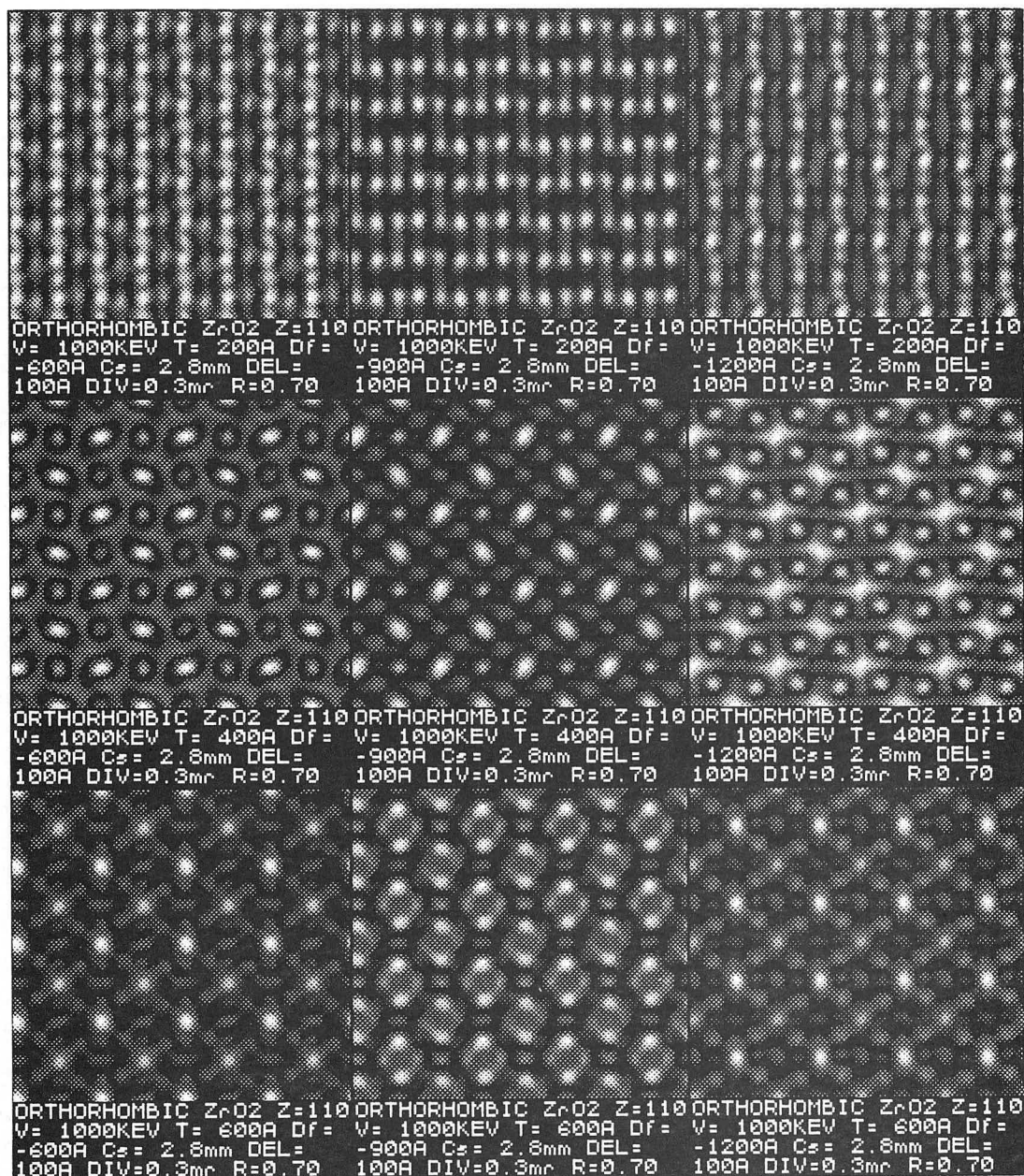
Fig. 19. Through-focus series of NCEMSS images of [110] orthorhombic ZrO_2 simulated under ARM conditions and output to the Gould IP9527 image processor, then screen copied to the LaserWriter.